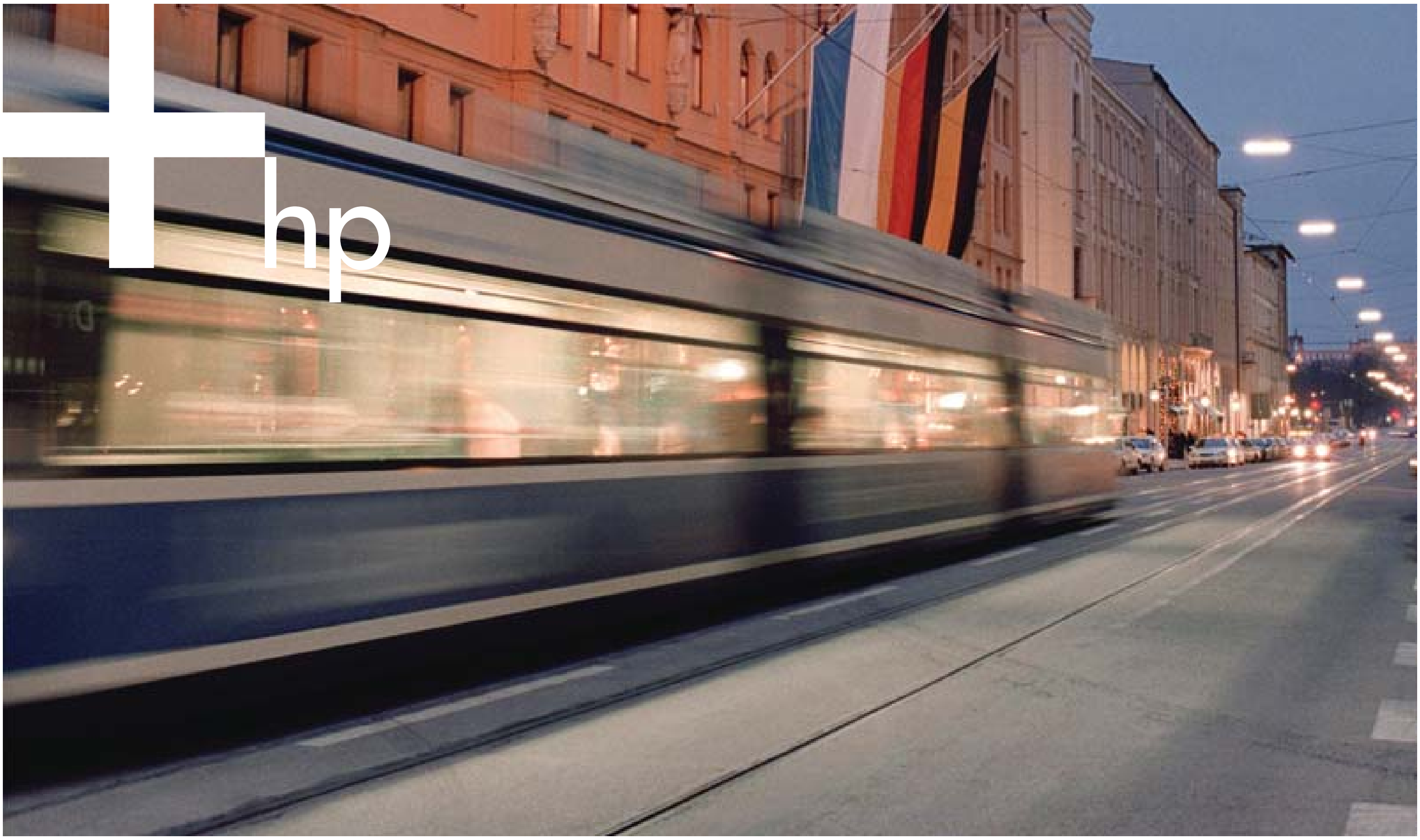
HP Autostore Server Software

Version 3.02



Support Guide

AutoStore version 3.02 support guide

Support guide

Copyright

The information contained herein is subject to change without notice.

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Edition 3, 08/2005

Trademarks

Table of contents

1 Overview

2 Setup and Installation

3 Using AutoStore for the first time

4 AutoStore processes

5 Capture components

6 Process components

7 Route components

8 AutoStore templates

9 Secured transport

10 Problem solving

1 Overview

Every company in business today faces the challenges of paper management. How does a company efficiently create, store, and distribute millions of business-critical, paper-based documents? Many companies are answering this question by storing paper-based information online and making this information available to employees by using client software programs. Most are finding this method to be a faster and less expensive solution for document distribution than sending stacks of computer-generated output to multiple recipients.

This document presents an overview of the AutoStore Server software (version 3.02) and provides administration instructions, installation procedures, usage information, and troubleshooting guidelines. Administrators can also use this document to understand the technical architecture, features, and capabilities of AutoStore.

See the glossary at the end of this document for information about terms and acronyms that might be unfamiliar.

Introduction to AutoStore

AutoStore is a middleware product designed to capture, process, and route business information. AutoStore runs as a Microsoft® Windows® service that is independent of any messaging system or database programs. AutoStore simply acts as a portal through which documents can travel between a device (for example, an HP digital sender) and software programs.

What does AutoStore do?

By using AutoStore, you can directly link digitized documents that were created by using on-ramp devices, such as an HP digital sender, to line-of-business software programs, such as Microsoft SharePoint Portal Server, Microsoft Exchange, Lotus Notes, and OLEDB database. By using AutoStore, you can also create workflows between on-ramp devices and other software programs.

AutoStore can be configured to process documents through various filters, such as optical character recognition (OCR), field mapping, and image cleanup. After a document is processed through one or more of these filters, it can be stored in a database, routed to an e-mail server, stored on a file transfer protocol (FTP) server, or sent to any number of available route destinations.

How does AutoStore work with digital senders and MFPs?

Use AutoStore to create custom-defined function keys for line-of-business programs, such as a Microsoft Exchange public folder database, a Lotus Notes database, or any other database. After you define the function keys and configure AutoStore, simply press a key on the digital sender control panel or the multifunction peripheral (MFP) control panel to scan a document. AutoStore routes the digitized images to folders and a variety of document management systems to meet your document-management requirements.

AutoStore overview

AutoStore contains four major components: the License Manager, the Process Designer, the Service Manager, and the Status Monitor.

- **AutoStore License Manager**. The License Manager grants licenses for using AutoStore and AutoStore components.

  Use the License Manager to update a license, transfer it to another computer, load the license onto a computer, or generate the license itself. For more information about the License Manager, see the detailed AutoStore License Manager description following this section.
- **AutoStore Process Designer**. AutoStore Process Designer is a graphical design tool that you can use to assemble the appropriate components to form a content workflow process. The AutoStore Process Designer is the main tool that you use to design and edit your configurations in AutoStore.
- **AutoStore Service Manager**. Use the Service Manager to start an AutoStore configuration file. A single configuration file can contain one or multiple processes.
- **AutoStore Status Monitor**. Use the Status Monitor to monitor status messages on a local or remote server that is powered by AutoStore.

AutoStore License Manager

The License Manager grants licenses for using AutoStore and AutoStore components. Use the License Manager to update a license, transfer it to another computer, load the license onto a computer, or generate the license itself.

NOTE AutoStore is licensed per computer. The licenses are directly related to the unique serial numbers that are reported by the AutoStore License Manager.

You can also select **Web License** and **Check for software update** to use those features. The following options are available when you use the License Manager:

Update. Type the static unlock key so that you can update your license.

Transfer. Transfer a licensed version of AutoStore to another server. The **Transfer** button generates two unique numbers that can be used to activate another AutoStore server. Note that the licensing on the server from which the AutoStore license is transferred expires when you transfer that license to another server.

Extend. Extend the evaluation dates by typing unique serial numbers that your reseller provides.

1. Open the AutoStore License Manager.

2. Click **Generate**.

3. Save the license information as a .TXT file.

4. Contact HP to submit the .TXT file and request an unlock extension key.

NOTE Extending the evaluation period is granted at the discretion of Hewlett-Packard. The evaluation period for the software cannot be extended until the evaluation period expires. You can only extend a license once.

Generate. Generate a serial number file that is unique to your AutoStore server and that you can send to HP to license AutoStore when the AutoStore server does not have access to the Internet. More information about licensing an AutoStore server when the AutoStore server does not have access to the Internet is provided later in this section. If the AutoStore server has access to the Internet, the **Web License** option should be used instead of the **Generate** option.

Load. Load the license file that HP support sent when you used the **Generate** option so that you can license the necessary components.

Bulk Transfer. Transfer licenses from the source computer to the destination computer.

1. On the destination computer, open the AutoStore License Manager and click **Generate**.

2. On the source computer, click **Transfer bulk**. Select the license file that was generated in step 1. Click **Open**. Specify the location of the license file that you are creating. Type the license file name, and then click **Save**.

NOTE Only components that are not already licensed on the destination computer can be transferred from the source.

3. On the destination computer, load the license file that you generated in step 2.

- **Web License.** Use this automated process to post your serial numbers through a Web portal and activate your current version of AutoStore. This process requires Internet access. Note that corporate firewalls can interfere with this operation.
- **Check for software update.** Check for software updates to make sure that the component that you are using is the most current version. For more information, refer to the Update Service.

  The Update Service is a web-based service that is provided to help you keep your software up to date. The Update Service allows you to view a list of available updates and to get information about the updates. Type your password to download and install the updates. Click **Show Updates** on the Update Server home page to see a list of the updates that are available for AutoStore. The updates are listed with their description and size. Select the update that you want and follow the wizard instructions to download and install the update.

You can use the following License Manager features to activate AutoStore components: Web License, No Internet access, and Transfer.

Web License. To obtain a Web license, the AutoStore server must have complete Internet access.

1. Open the AutoStore License Manager.

2. Click **Web License** to view the Web browser window.

3. Follow the instructions and provide all of the required information.

NOTE You must have a valid e-mail address.

4. Save the license file that you receive to your server hard disk.

5. In AutoStore License Manager, click **Load**.

6. Open the activated license file that you saved in step 4.

No Internet access. You can license an AutoStore server that does not have Internet access.

1. Open the AutoStore License Manager.

2. Click **Generate**.

3. Save the license information as a .TXT file.

4. Transfer the .TXT file to a computer that has Internet access. On the computer that has Internet access, open an Internet browser window and type `http://licensing.hp.com/`. Follow all of the instructions and provide all of the required information.

5. Save the license file to your server hard disk.

6. In AutoStore License Manager, click **Load**.

7. Open the activated license file that you saved in step 5.

Transfer license. You can transfer a license to a different server.

NOTE After you transfer a license to another server, you cannot transfer that license back to the original server.

1. Install the AutoStore software on a different server.

2. Open the AutoStore License Manager on the server on which you just installed the software.

3. Write down the eight-digit MFP component number (unlock key) that is located in the SN column.

4. On the original server, click the MFP component.

5. Click **Transfer**.

6. Type the eight-digit number (unlock key) that you wrote down from the new server in step 3. A warning window appears that explains that the licenses for the original server will expire immediately.

7. Write down the eight-digit unlock key that you will need for the new server.

8. On the new server, click the MFP component.

9. Click **Update**.

10. Type the eight-digit unlock key that you wrote down from the original server in step 7.

AutoStore Process Designer (APD)

Use the APD to create processes with an easy-to-use graphical user interface (GUI). The APD offers the following features:

- The ability to view all of the attributes related to the process you are creating
- The flexibility to design and implement processes using any number of AutoStore components
- Visual integration for all third-party or custom-built components
- Support for advanced servers such as SharePoint Portal, Oracle IFS, Domino.DOC, and so on

For more information about using the APD and creating processes, see the "AutoStore processes" section in this document.

AutoStore Service Manager

Use the AutoStore Service Manager to start an AutoStore configuration file. AutoStore must be running as a service on a PC with the appropriate operating system before your process can work correctly.

1. Take one of the following actions to start AutoStore:

   On the computer desktop, click **Start**, select **Programs**, select **AutoStore**, and then click **AutoStore Service Manager**.

   On the computer desktop, click **Start**, select **Settings**, select **Control Panel**, and then click **AutoStore Service Manager**.

   The **AutoStore** dialog box appears.

2. Type the field names from the following table, and then click **Apply**.

Field name	Description
Status	This is a display-only field that shows the status of the service. The default status of the service is **Stopped**.
AutoStore Script	This field contains the configuration file name. The AutoStore Process Designer generates this file when you save your AutoStore parameters. The file has a .CFG file extension. Either type the address or click the ellipses button to select a configuration file in the Windows Explorer view. A drop-down box shows recently-used scripts.
Startup Type	Using this field, you can select the startup type for the service. **Automatic** restarts the service automatically if the server is restarted. **Manual** requires the service to be restarted manually every time the server is restarted. **Disabled** marks the service as disabled so that it cannot be started.
Log On NT As	This field contains three additional fields, **Account**, **Password**, and **Confirm Password**. ● **Account.** Type the user name of the local computer where AutoStore is installed. The default is **LocalSystem**. If the folder where you store processed files is located on a remote computer on the network, you must provide proper credentials (at a minimum, provide write permissions) to the folder. Type the **Domain Name** followed by the **Account Name** (DOMAIN NAME\NT USER ID). ● **Password.** Specify a valid password. The default password for **LocalSystem** is blank. Do not type anything in the space provided. ● **Confirm Password.** Type the same password that you specified in the **Password** field. If you left it blank, then leave this field blank.

AutoStore Status Monitor

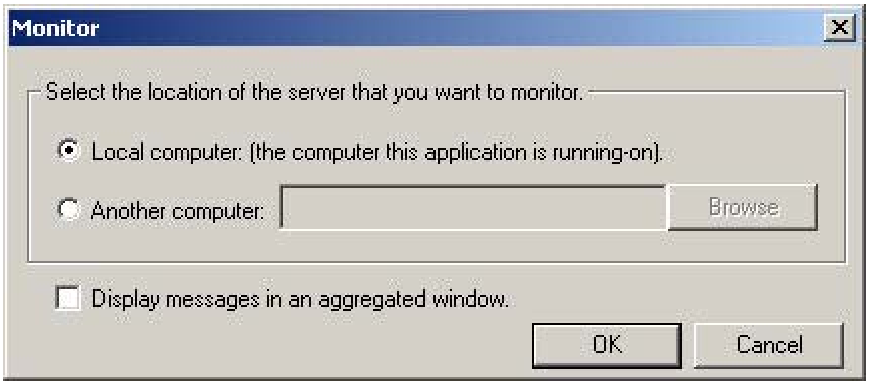
The AutoStore Status Monitor shows real-time status messages that are associated with all of the active processes on a server that is running AutoStore. Monitoring these status messages helps you produce the appropriate result for a given process. It can also help predict and identify the sources of any potential system problems. Status Monitor messages contain the following information.

- **Type.** The type of status message. This can be one of the following types:
  - **Error.** Error types indicate significant problems that you should know about. Error events usually indicate a loss of functionality or data.
  - **Warning.** Warning types indicate problems that are not immediately significant, but that might indicate conditions that could cause future problems.
  - **Information.** Information types provide messages about your operations.
- **Message.** The message text that is associated with an event.
- **Time.** The time of the event on the server.

The Status Monitor also creates a log file of everything that appears in the following location: %SYSTEMROOT%:\%WINDOWSFOLDER%\SYSTEM32\LOG\. For example, C:\WINDOWS\SYSTEM32\LOG. The log file is not affected by the 512 limit.

The following features are available on the Status Monitor toolbar.

- **Clear.** Click the **Clear** icon to clear any existing status messages that appear in the current tab window.
- **Connect.** Click the connect icon to open the **Monitor** dialog box.


Monitor

Select the location of the server that you want to monitor.

● Local computer: (the computer this application is running-on).

○ Another computer:

□ Display messages in an aggregated window.

  Local computer is the default selection. To connect to a remote computer, click **Browse** to select a computer or specify the remote server IP address. This dialog box also allows you to select whether to aggregate or separate the process messages. If messages are aggregated, then all of the messages from all of the active processes appear within one tab window. Otherwise, a separate tab window is created for each process.

- **Disconnect.** Click the **Disconnect** icon to close the current connection to the server that is being monitored.
- **Save.** Click the **Save** icon to save the status messages in the current tab window to a .TXT file. Use this feature to save the messages and then send them to support personnel or for analysis.

- **Stop.** Click this icon to stop the message stream from showing any additional messages. Use this feature when you are working with a high-volume-processing server that generates a large number of status messages.
- **Continue.** Click the **Continue** icon to resume showing status messages if you have previously stopped the message stream.
- **Copy.** Click this icon to copy any highlighted text to the clipboard so that you can paste it to another location.
- **Clear.** Click the **Clear** icon to clear any existing messages from the current tab window.
- **Find.** Click this icon to search for characters or words that you want to find in the status messages. Type the characters or words that you want to find, and then click **Find what**. Click **Find Next** to begin searching.
- **Print.** Click this icon to print the currently selected tab window. Use the `Print` command to print selected log entries.
- **Zoom.** Click the **Zoom** icon to adjust the font size of the status message.
- **Type.** Click this icon to filter the current status messages based on message type. Select all message types or select a particular message type to use as the filter.
- **Max Entries.** Click the **Max Entries** icon to control the number of entries that appear in a tab window before the messages wrap. This feature only affects the number of message entries in the Status Monitor; it does not affect server log files.

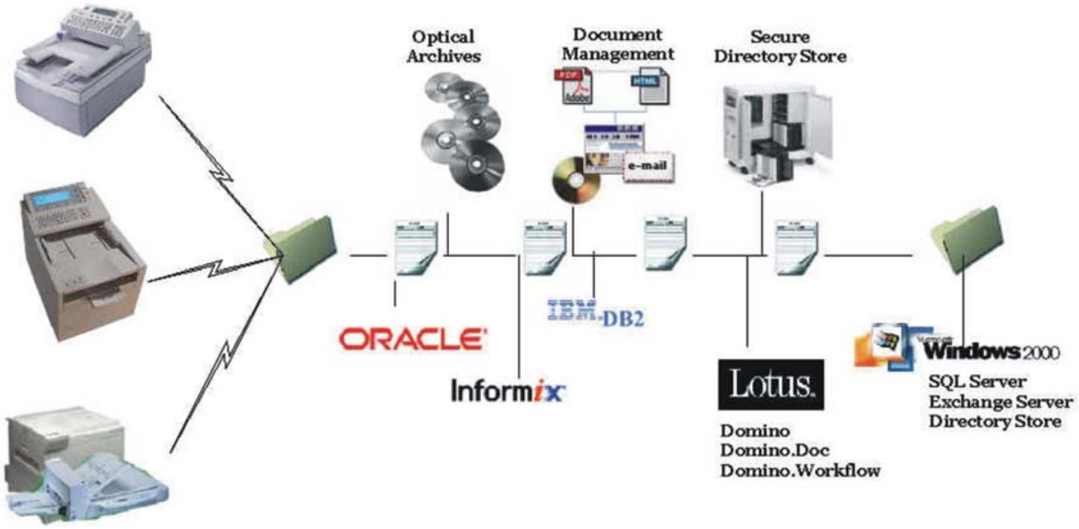
When the status messages exceed the column width of the Status Monitor, the information is denoted by an ellipses (...). You can select the message by clicking on it, then copy and paste it to any text editor to view the entire message, even the part that exceeded the width of the Status Monitor.

NOTE

If you place the cursor over a status message that contains additional information (exceeds the width of the Status Monitor), a tool-tip yellow line appears. The tool tip might contain the additional information, but in some cases the status message has more information than the tool tip can provide. In these cases, copy and paste the message into a text editor to view the entire message.

AutoStore architecture

The following illustration shows how AutoStore captures documents from HP digital senders and MFPs, and routes documents to various storage locations after processing.



AutoStore consists of three distinct layers called **Capture**, **Process**, and **Route**. Each component within these layers works with AutoStore as a separate multi-threaded process.

New AutoStore features

AutoStore provides a strong development framework for software programs. The following are the latest enhancements:

- AutoStore provides software-development framework support for all types of components. You can easily develop your own components and use them within an AutoStore process.
- AutoStore offers support for expanded function keys and prompts. AutoStore is fully integrated with function-key support for prompt fields on the latest HP digital senders and MFPs.
- AutoStore offers full support for the Microsoft SharePoint Portal Server. It provides two separate components to connect data to the Microsoft SharePoint Portal:
  - Use the SharePoint Portal 2003 Route component to store images and data in SharePoint Portal.
  - Use the SharePoint Portal 2003 Process component to store files in SharePoint Portal, and then create links to those files.

2 Setup and Installation

The setup and installation process consists of the following steps. Look for detailed instructions in the sections that follow.

- **Step 1: Uninstall any existing AutoStore software.**

NOTE Skip this step if you have never installed AutoStore.

- **Step 2: Install the latest AutoStore Server software (version 3.02), and then use the Check for Software Updates button in the License Manager to upgrade to the latest version..** Please read the Important Notes document before you install the AutoStore Server software. Run the AutoStore installation file (HPAUTOSTORE.EXE) and follow the instructions on the computer screen.
- **Step 3: Configure devices and update components.** Follow the component installation and upgrade procedures to complete the system configuration, and to convert any existing configuration files to the new AutoStore configuration file format.

Before you begin

If you are using AutoStore for any supported databases or line-of-business programs, make sure that the client you use to gain access to this database or line-of-business application is installed and configured on a different server *before* you install and configure the AutoStore software. For example, if you want AutoStore to work with Microsoft SharePoint Portal server, first make sure that Microsoft SharePoint Portal Server is fully installed and configured.

If you are evaluating AutoStore, you must purchase and install a license key within 60 days of installation or before 2000 documents are processed, whichever occurs first. If you do not install the license key within 60 days or before 2000 documents are processed, AutoStore is no longer available. Any adjustment to computer dates or registries causes the AutoStore licenses to expire automatically.

System requirements

Make sure that your system meets the following minimum requirements for installation:

- Windows 2000 Server with Service Pack 4 or Windows Server 2003
- Intel Pentium® III processor, 600 megahertz (MHz) or faster
- Support database engine client, such as Microsoft Exchange 5.5 with Service Pack 3, Lotus Notes 4.6.*X*, or any other supported client software program (if you plan to incorporate these items with AutoStore)
- At least 512 megabytes (MB) of dedicated random-access memory (RAM)
- At least 512 MB of available hard-disk space
- For the HP LaserJet 4100mfp, firmware version 03.801.1 or later
- For the HP LaserJet 9000mfp, firmware version 03.801.1 or later
- HP LaserJet 4345mfp Series, firmware version 09.022.3 or later
- HP LaserJet 9050mfp and HP LaserJet 9040mfp, firmware version 08.021.7 or later
- HP LaserJet Color 9500mfp, firmware version 08.021.7 or later
- HP LaserJet 9055mfp and HP LaserJet 9065mfp, firmware version 07.004.0 or later
- HP 9200C Digital Sender, firmware version 09.022.1 or later
- HP Jetdirect Card J6057A or newer

NOTE To obtain firmware information, print a configuration page from the embedded Web server or the device control panel. See the device Help for instructions.

NOTE To obtain HP Jetdirect Card information, use the embedded Web server. In the Web browser address line, type `http://<deviceaddress>`, and then press Return. In the embedded Web server, click the **Networking** tab. Click **Configuration Page** and look for the model number.

Supported devices

The AutoStore Server software supports the following HP devices:

- HP 9100C Digital Sender
- HP LaserJet 4100mfp and HP LaserJet 4101mfp
- HP LaserJet 9000mfp
- HP LaserJet 9055mfp and HP LaserJet 9065mfp
- HP LaserJet 4345mfp Series
- HP LaserJet 9200C Digital Sender
- HP Color LaserJet 9500mfp
- HP LaserJet 9050mfp and HP LaserJet 9040mfp

Step 1: Uninstall any existing AutoStore software

Complete the following procedure to remove the existing AutoStore Server software.

To uninstall the AutoStore Server software

1. Stop and close all AutoStore-related programs, including the following programs:
    - AutoStore Service
    - Status Monitor
    - AutoStore Process Designer
    - AutoStore License Manager
2. Click **Start**, select **Settings**, select **Control Panel**, and then click **Add or Remove Programs**. In the **Add or Remove Programs** dialog box, click **AutoStore**, and then click **Remove**.
3. Close the **Add or Remove Programs** dialog box.
4. Shut down and restart the computer.
5. In Windows Explorer, browse to the AutoStore installation folder. (Depending on which version of AutoStore you have installed, the default location is either C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE or C:\PROGRAM FILES\NSI \AUTOSTORE.) Delete the AutoStore installation folder.

Step 2: Install the AutoStore Server software

Use the following procedure to install the software.

NOTE If you plan to install the Lotus Notes/Domino Server, you must install the Lotus Notes client software on the server before you install the AutoStore Server software. If you do not want AutoStore to work with the Lotus Notes/Domino Server, make sure that the Lotus Notes/ Domino Server option is *not* selected in the Route component list during installation.

If you do not want to use AutoStore with SharePoint Portal 2003, make sure that the SharePoint Portal 2003 option is *not* selected in the component list during installation.

To install the AutoStore Server software

1. Double-click the HPAUTOSTORE.EXE file and follow the instructions on the computer screen.

Go to Step 3: Configure devices and update components for additional component-specific information. You must complete the additional procedures for any devices that you plan to use with AutoStore, such as MFPs or digital senders.

If you have previously used AutoStore R1.00, follow the instructions in Step 3: Configure devices and update components to complete the component-specific installation, and to convert any existing configuration file to complete your installation.

Typically, When you insert the HP AutoStore disk, installation begins automatically if you have enabled autorun. If you insert the HP AutoStore disk and installation does not begin automatically, browse to find the HPAUTOSTORE.EXE file. Double-click the HPAUTOSTORE.EXE file and follow the instructions on the computer screen.

Step 3: Configure devices and update components

Some devices and components require additional installation or configuration. Complete the procedures that apply to the devices and components that you plan to use with AutoStore.

Uninstalling Chai .JAR files on the device

If you have previously installed Chai .JAR files on your MFP device, follow these steps to remove them *before* you install the new Chai .JAR files.

1. Find the Internet protocol (IP) address of the printer by using one of these methods:

   Ping the printer name on the MS-DOS command screen.

   Print the Configuration pages (click **Menu**, click **Information**, and then click **Print Configuration**).

2. Type the following universal resource locator (URL) in the address line of a Web browser: `http://PrinterIPAddress/hp/device/this.loader`.
3. In the **Select Package** dialog box, select the check box next to HP_AUTOSTORE_4100_9000_XXX.JAR, and then click **Remove Selected Packages**.
4. Review the Confirmation page to make sure that the package is uninstalled correctly.
5. In the **Select Package** dialog box, select the check box next to HP_LASERJET_DYNAMICMENUS_XXX_X.JAR, and then click **Remove Selected Packages**.
6. In the **Select Package** dialog box, select the check box next to HP_LASERJET_ADDRESSBOOK_XXX_X.JAR, and then click **Remove Selected Packages**.
7. Review the Confirmation page to make sure that the package is uninstalled correctly.

Installing Chai.JAR files on the device

When you installed the AutoStore Server software, the Chai .JAR (JAVA archive) files that are required for each MFP or digital sender were saved in the CHAIJARS folder within the AutoStore program directory. Use one of the following methods to install the Chai .JAR files on the device. Each method provides information about how to install the AutoStore .JAR files on an MFP. Use the method that is best suited for your environment.

The .JAR file format is used to bundle all of the components that an HP Chai applet requires. The .JAR file format simplifies the process of downloading applets because all of the components (.CLASS files, images, sounds, and so on) can be packaged into a single file. The .JAR file format also supports data compression, which further decreases download time. Chai is the HP-proprietary JAVA for MFP devices. HP Chai is functioning within the MFP device, and the workflow server is designed to work with the Chai-enabled HP MFPs.

Method 1: To install Chai .JAR files by using HP Web Jetadmin (HP LaserJet 4100mfp and HP LaserJet 9000mfp, HP LaserJet 4345mfp, HP LaserJet 9050mfp and HP LaserJet 9040mfp, HP LaserJet Color 9500mfp, and HP 9200C Digital Sender)

1. Copy all of the files from the CHAIJARS subdirectory in the AutoStore installation directory (C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE\CHAIJARS) to the Web Jetadmin plugin directory (C:\PROGRAM FILES\HP WEB JETADMIN\DOC \PLUGINS\HPJDAM\JARS).

2. Use a Web browser to go to the HP Web Jetadmin home page at the following address: http://<hostname>:8000.

3. Open the Device Application Manager by using one of the following methods:

   On the printer Device Status page, select **Application Manager** from the drop-down menu on the content toolbar.

   On the Device Management page, click **Device Lists**, click **All Devices**, and then select the device(s) from the list.

   On the **Device Tools** drop-down menu, click **Application Manager**.

   On the Device Management page, click **Device Groups**, click the group name, and then select the device(s) from the list.

   On the **Device Tools** drop-down menu, click **Application Manager**.

4. On the Install page, select the **HP AutoStore for mfp** device software program, and then click **Install**. Your printer automatically reboots.

NOTE The Installation Results page shows the results of the installation process for an individual device or devices in a group. Use this page to verify that the Device Application Manager successfully installed the device software program on an individual device or devices in a group.

5. Configure AutoStore for the MFP device or device group. On the Device Management page, click **Device Lists**, click **All Devices**, and then select the device or device group from the list.

6. On the **Device Tools** drop-down menu, click **Configuration**. The default configuration category is **Device**.

7. Specify the AutoStore configuration attributes depending on your system environment. Set the **Port** attribute to 3232.

Method 2: To install Chai .JAR files by using the PJL file and FTP from the Command prompt (HP LaserJet 4100mfp and HP LaserJet 9000mfp, HP LaserJet 4345mfp, HP LaserJet 9050mfp and HP LaserJet 9040mfp, HP LaserJet Color 9500mfp, and HP 9200C Digital Sender)

NOTE The printer job language (PJL) file installation replaces all existing Chai installations. If you have previously installed any other Chai .JAR files for any other program, those installations will be overwritten.

1. Open an MS-DOS command window on your computer.
2. Type `FTP <TCP/IP PRINTER ADDRESS>` (for example, if the transmission control protocol/Internet protocol [TCP/IP] address is 192.168.0.90, type `FTP 192.168.0.90`. Press Enter on the keyboard.
3. When prompted for a user name and password, press Enter for each prompt (no user, no password).
4. Type `bin`, and then press **Enter** to switch to binary mode.
5. Type: `put <path>` where `<path>` is the location of the HP_AUTOSTORE_4100_9000.XXXX.PJL file. For example, type: `put C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE\CHAIJARS \HP AUTOSTORE_4100_9000.XXXX.PJL` and then press Enter on the keyboard. The printer automatically reboots.
6. After the transfer has been completed, you can type `BYE` at the command prompt and then press Enter to exit the FTP session.

Method 3: To install Chai .JAR files by using the PJL file and Windows Explorer (HP LaserJet 4100mfp and HP LaserJet 9000mfp, HP LaserJet 4345mfp, HP LaserJet 9050mfp and HP LaserJet 9040mfp, HP LaserJet Color 9500mfp, and HP 9200C Digital Sender)

NOTE The PJL file installation replaces all existing Chai .JAR file installations. If you have previously installed any other Chai .JAR files for any other program, those installations will be overwritten.

1. To enable your browser for FTP open Windows Explorer, click **Tools**, click **Internet Options**, and then click **Advanced**. Scroll to the **Browsing** section and select **Enable Folder view** for FTP sites.
2. Open a Windows Explorer window and type `FTP://<MFP IP Address>`. When the connection is made a Port1 folder appears on the mfp.
3. Navigate to the local folder that contains the Chai .JAR files and drag and drop the .PJL file to the Port1 folder on the mfp.

Uninstalling Chai .JAR files on the device (HP LaserJet 9055mfp or 9065mfp)

If you have previously installed Chai .JAR files on your MFP device, follow these steps to remove them *before* you install the new Chai .JAR files.

1. Find the IP address of the printer by using one of these methods:

   Ping the printer name on the MS-DOS command screen.

   Print the Configuration pages (click **Menu**, click **Information**, and then click **Print Configuration**).

2. Type the following URL in the address line of a Web browser: `http://PrinterIPAddress/hp/device/this.loader`.

3. In the **Reloadable Packages** dialog box, select the check box next to **AutoStoreDC**, and then click **Remove Selected Packages**.

4. Review the Confirmation page to make sure that the package is uninstalled correctly.

Installing Chai .JAR files on the device (HP LaserJet 9055/9065mfp)

When you installed the AutoStore Server software, the Chai .JAR (JAVA archive) files that are required for each MFP or digital sender were saved in the CHAIJARS folder within the AutoStore program directory. Use one of the following methods to install the Chai .JAR files on the device. Each method provides information about how to install the AutoStore .JAR files on an MFP. Use the method that is best suited for your environment.

The .JAR file format is used to bundle all of the components that an HP Chai applet requires. The .JAR file format simplifies the process of downloading applets because all of the components (.CLASS files, images, sounds, and so on) can be packaged into a single file. The .JAR file format also supports data compression, which further decreases download time. Chai is the HP-proprietary JAVA for MFP devices. HP Chai is functioning within the MFP device, and the workflow server is designed to work with the Chai menu-enabled HP MFPs.

Method 1: To install Chai .JAR files by using the HP Web Jetadmin (HP LaserJet 9055mfp and 9065mfp)

1. Copy all of the files from the ChaiJars subdirectory in the AutoStore installation directory (C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE\CHAIJARS) to the Web Jetadmin plugin directory (C:\PROGRAM FILES\HP WEB JETADMIN\DOC\PLUGINS\HPJDAM\JARS).

2. Use a Web browser to go to the HP Web Jetadmin home page at the following address: http://<hostname>:8000.

3. Open the Device Application Manager by using one of these methods:

   On the printer Device Status page, select **Application Manager** from the drop-down menu on the content toolbar.

   On the Device Management page, click **Device Lists**, click **All Devices**, and then select the device(s) (HP LaserJet 9055mfp or HP LaserJet 9065mfp)from the list.

   On the **Device Tools** drop-down menu, click **Application Manager**.

4. On the Install page, select **HP AutoStore for 9055mfp/9065mfp** device application for this device type, and then click **Install**. The printer automatically reboots.

5. Configure AutoStore for the MFP device or device group. On the Device Management page, click **Device Lists**, click **All Devices**, and then select the device (HP LaserJet 9055mfp or HP LaserJet 9065mfp) or device group from the list.

6. On the **Device Tools** drop-down menu, click **Configuration**. The default configuration category is **Device**.

7. Specify the AutoStore configuration attributes depending on your system environment. Set the **Port** attribute to 3434.

Method 2: To install Chai .JAR files by using the embedded Web server (HP LaserJet 9055mfp and 9065mfp)

1. Obtain the IP address of the device by using one of these methods:

   Ping the device name on the MS-DOS command screen.

   Print the Configuration pages (click **Menu**, click **Information**, and then click **Print Configuration**).

2. Type the following URL in the address line in a Web browser: `http://PrinterIPAddress/hp/device/this.loader`.

3. Click **Browse**, which is next to **Enter Package File**.

4. Select the .JAR file, HP_AUTOSTORE_9055_9065_XL.XXXX.JAR, in the C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE\CHAIJARS directory and click **Open**.

5. Click **Load Package Now**. The printer automatically reboots.

6. Review the Confirmation page to make sure that the file was installed correctly.

7. Configure AutoStore for the MFP device or device group. Click the link to the AutoStore configuration page.

8. Specify the AutoStore configuration attributes depending on your system environment. Set the **Port** attribute to 3434.

Method 3: To install Chai .JAR files by using the PJL file (HP LaserJet 9055mfp and 9065mfp)

NOTE The printer job language (PJL) file installation replaces all existing Chai installations. If you have previously installed any other Chai .JAR files for any other program, those installations will be overwritten.

1. Open an MS-DOS command window on your computer.
2. Type `FTP <TCP/IP PRINTER ADDRESS>` (for example, if the transmission control protocol/Internet protocol [TCP/IP] address is 192.168.0.90, type `FTP 192.168.0.90`. Press Enter on the keyboard.
3. When prompted for a user name and password, press Enter for each prompt (no user, no password).
4. Type `bin`, and then press **Enter** to switch to binary mode.
5. Type: `put <path>` where `<path>` is the location of the HP_AUTOSTORE_9055_9065.XXXX.PJL file. For example, type: `put C:\PROGRAM FILES\HEWLETT-PACKARD\HP AUTOSTORE\CHAIJARS \HP AUTOSTORE_9055_9065.XXXX.PJL` and then press Enter on the keyboard. The printer automatically reboots.
6. After the transfer has been completed, you can type `BYE` at the command prompt and then press Enter to exit the FTP session.

Additional information

HP Digital Sending Product information and technical documentation is available on the HP Web site. Go to www.hp.com/go/mfp for information about MFPs and other MFP-related products.

3 Using AutoStore for the first time

Creating an AutoStore process involves several steps, including creating a configuration file, creating a workflow, configuring the components, running the process, and monitoring the results. All of these steps are discussed in this support guide.

Before you can run an AutoStore process, you must create a configuration file. Use this configuration file to start the AutoStore Service Manager and enter system information before you begin using AutoStore.

A configuration file (.CFG) contains a complete AutoStore process, including the process attributes. When you use the AutoStore Process Designer to create a process, the process is saved as a configuration file with the .CFG extension.

NOTE Before you can run your AutoStore process, you must configure the Capture, Process and Route components. Information about configuring the individual components is available in the individual Help files, and in the Capture, Process, and Route sections of this guide.

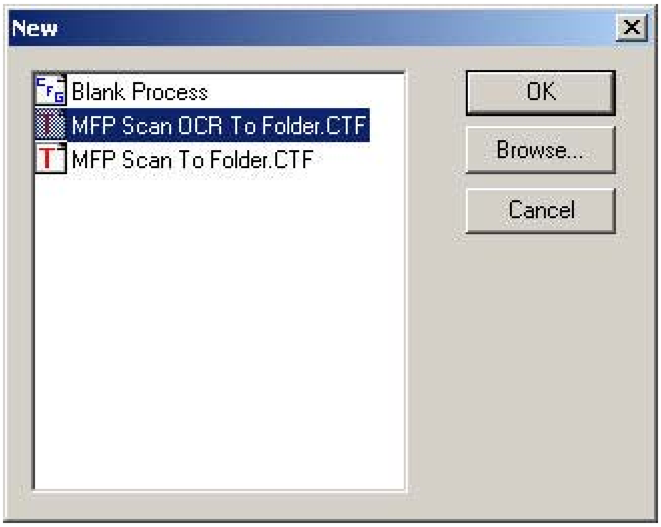
You can create a configuration file by using either a blank process or one of the AutoStore templates.

The following two sections show examples of creating a configuration file by using three processes: one process that uses an AutoStore template and two blank processes.

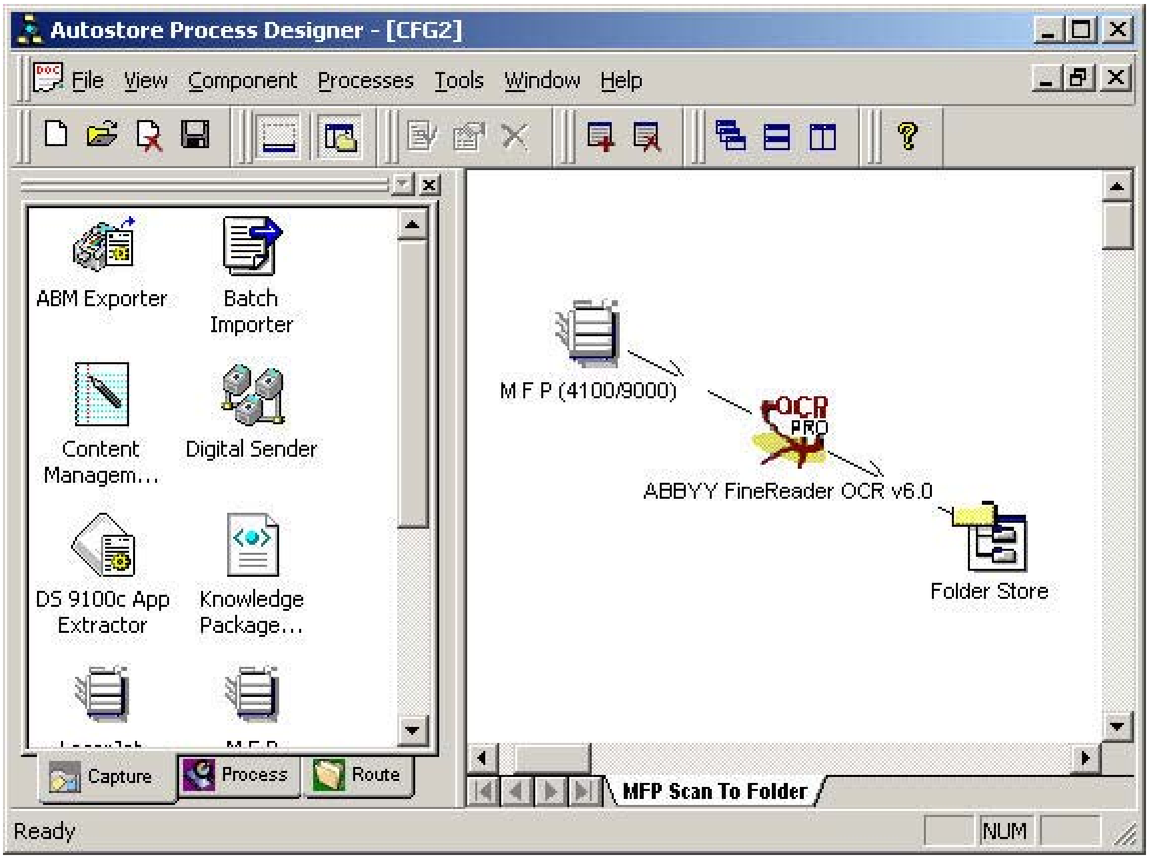
Creating a configuration file by using a template

A process template is a special kind of configuration file that provides basic tools for shaping an AutoStore process. Templates can contain one or more processes, process attributes and settings, and component attributes and settings. AutoStore provides two templates, which are installed with the AutoStore software. You can also create your own templates for storing components or settings that you want to reuse in other processes.

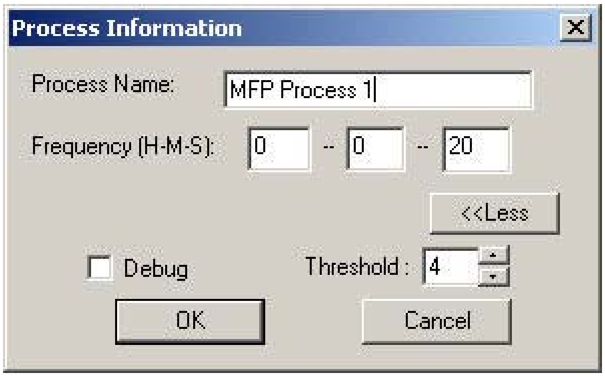
- In the AutoStore Process Designer, click **File**, and then click **New**.
- Select **MFP Scan OCR To Folder.CTF**, and then click **OK**.



- Double-click anywhere in the white space in the right-side pane.



- Set the frequency to 20 seconds and name the process MFP Process 1. Click **OK**.



Process Information

Process Name: MFP Process 1

Frequency (H-M-S): 0 -- 0 -- 20

<<Less

□ Debug

Threshold : 4

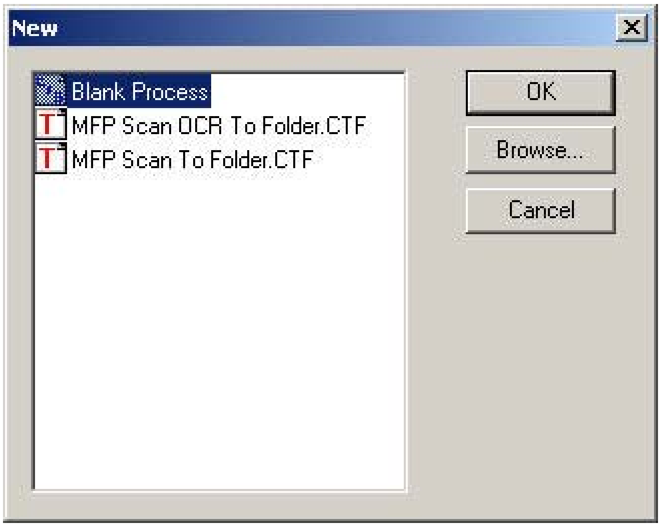
OK

Cancel

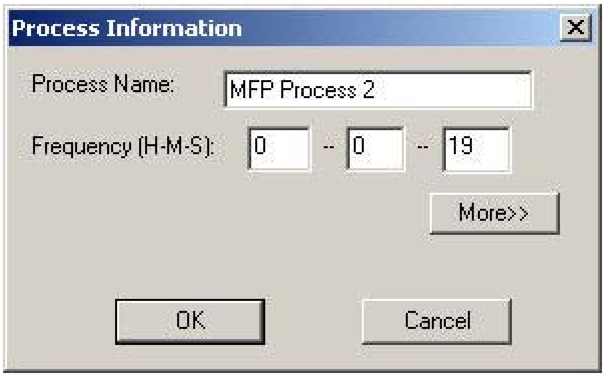
The frequency is specified in hours, minutes, and seconds and represents how often the Capture component communicates with the AutoStore Service Manager. The longer the frequency, the more time the process takes to run.

Creating a configuration file by using blank processes

- Open the AutoStore Process Designer. Click **Start**, select **Programs**, select **Hewlett-Packard**, select **HP AutoStore**, and then click **AutoStore Process Designer**.
- On the Process Designer toolbar, click **File**, and then click **New**.
- Select **Blank Process** in the **New** dialog box, and then click **OK**.



- The **Process Information** dialog box appears. This dialog box contains the process attributes. Set the frequency at 19 seconds. Name the process MFP Process 2. Click **OK**.



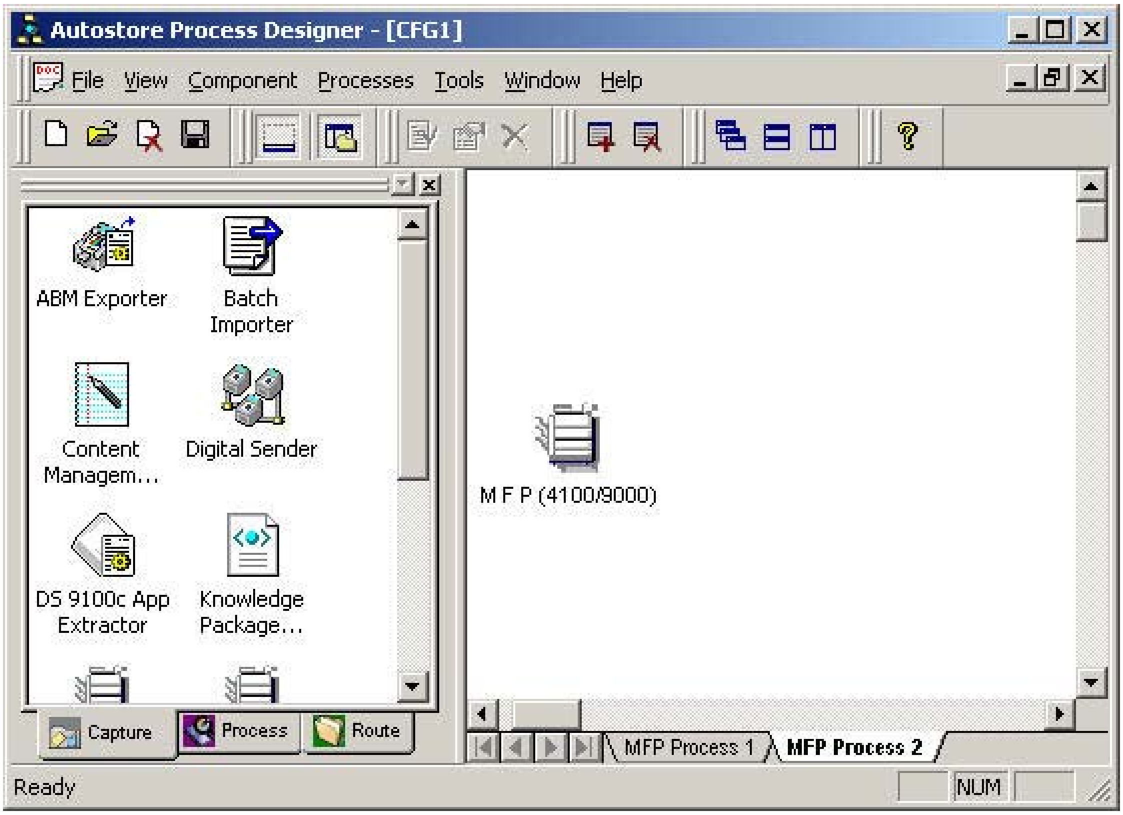
Process Name: MFP Process 2

Frequency (H-M-S): 0 -- 0 -- 19

If you click **More**, the following fields become available.

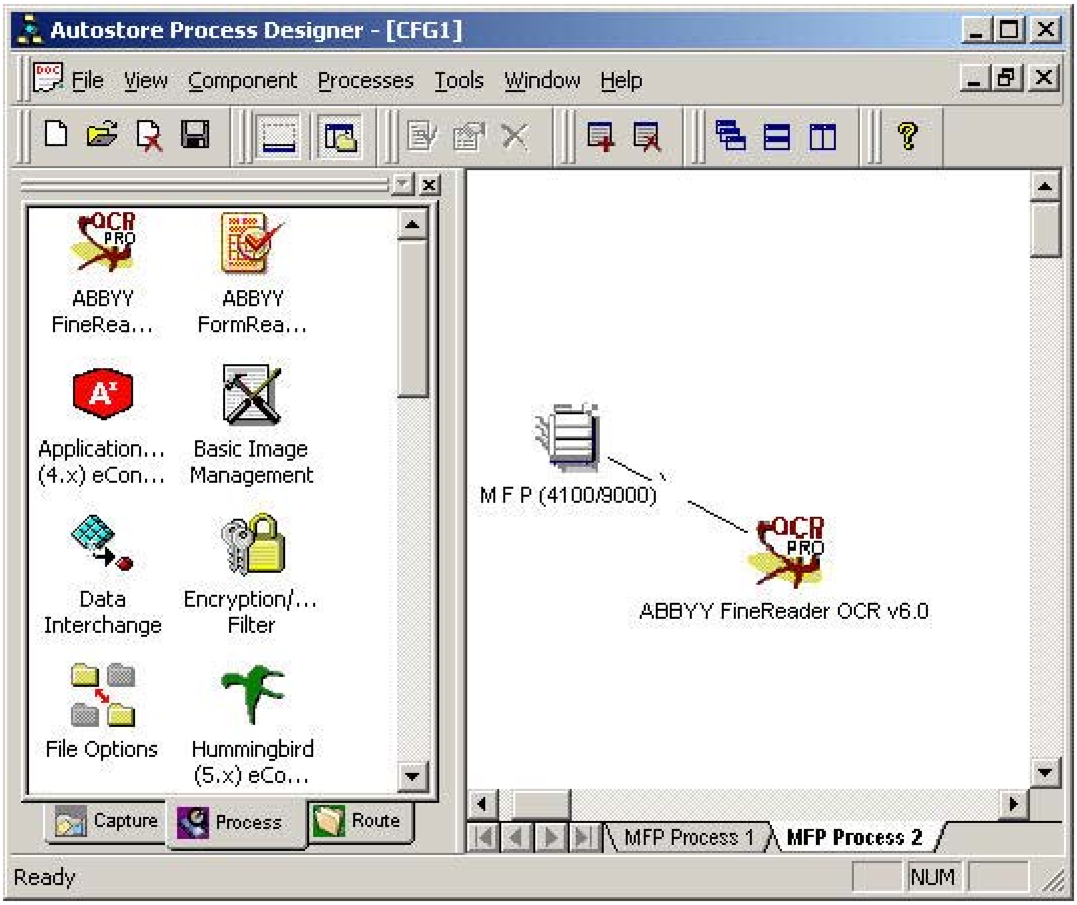
- The **Debug** check box is used only for support purposes. Only the AutoStore administrator should turn on this feature. When Debug is activated, the various components within AutoStore start creating log messages within the AutoStore Log file. These messages are designed to help the support staff take a closer look at the internal operations of an AutoStore server that is running in the background.
- In the **Threshold** field, use the spin box to specify a number between 0 and 10, with 0 representing the lowest level of logging and 10 representing the highest level of logging. The default is set to 4. When the Debug function is activated, the number of log messages from each component increases based on the Threshold level that you select. This operation requires free disk space and can affect system operations if it is not turned off. System performance and availability might be affected if the hard-disk storage is depleted.

- Drag and drop the MFP (4100/9000) Capture component from the component tray to the blank process in the right-side pane.

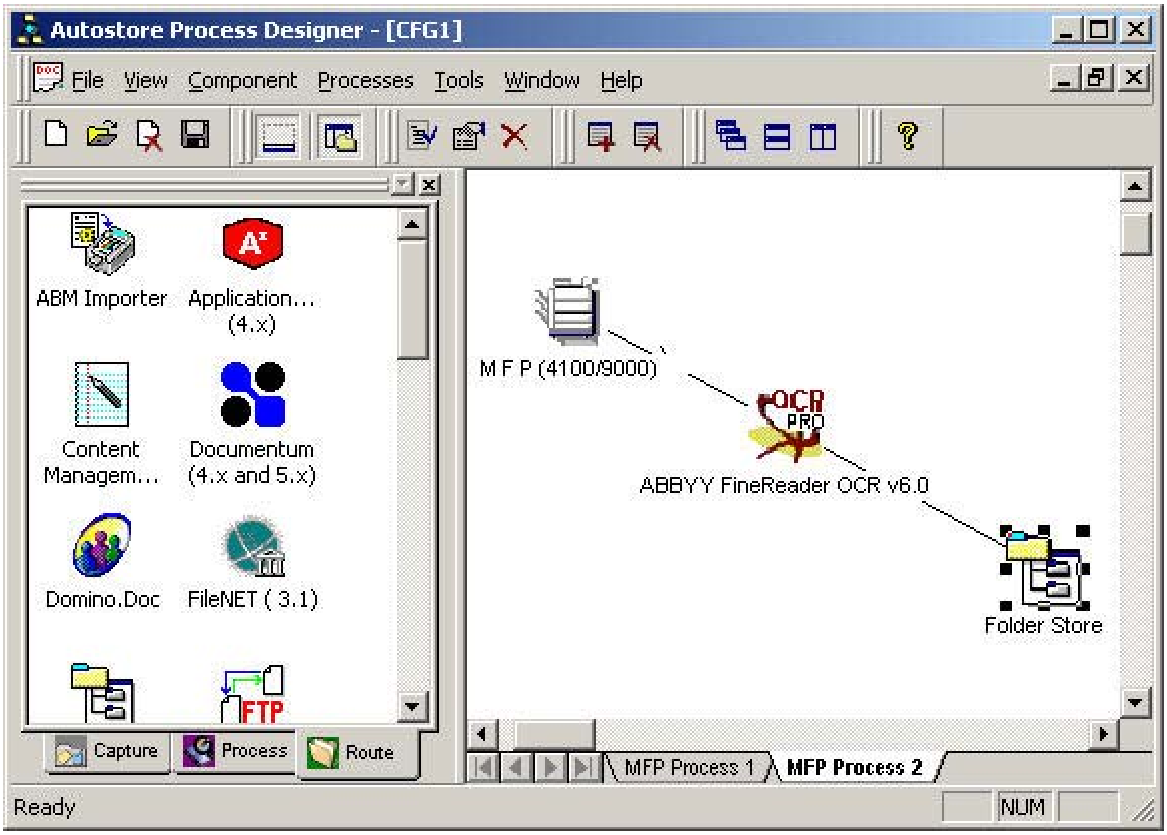


- Click the **Process** tab under the component tray.

- Use ABBYY FineReader OCR as the Process component. Drag and drop that component from the component tray to the right-side pane.



- Click the **Route** tab under the component tray.
- Drag and drop the Folder Store Route component to the right-side pane.



Now create the third and final process in the configuration file.

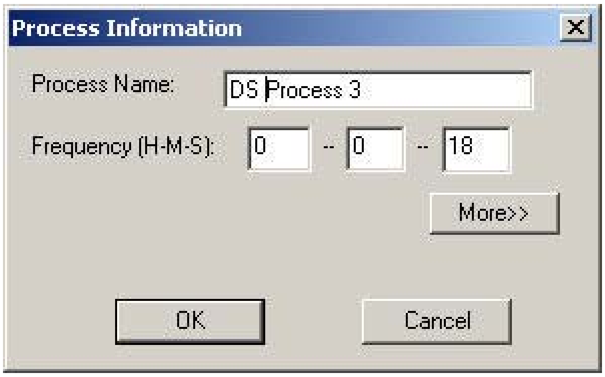
- On the Process Designer toolbar, click **File**, and then click **New**.
- Create a blank process by using one of the following methods.

  Select **Blank Process** in the **New** dialog box, and then click **OK**, or use one of the following:

  On the Process Designer toolbar, click **Processes**, and then click **New Process**. Select **Blank Process** in the **New** dialog box, and then click **OK**.

  Place your cursor in the white space in the right pane and right-click. Select **Blank Process** in the **New** dialog box, and then click **OK**.

- In the **Process Information** dialog box, set the frequency at 18 seconds. Name the process DS Process 3. Click **OK**.

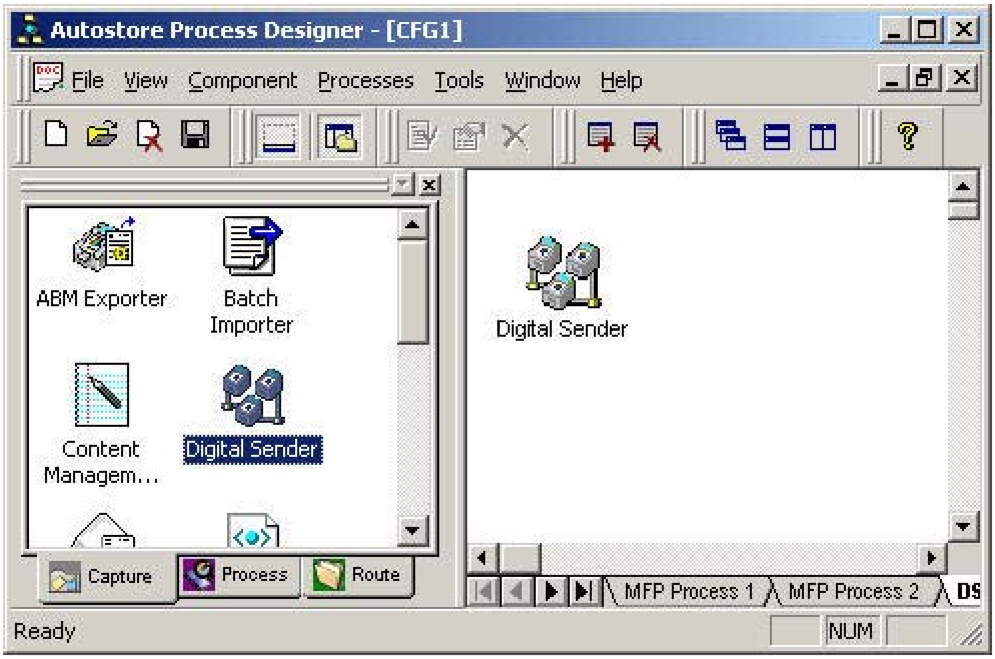


Process Information

Process Name: DS Process 3

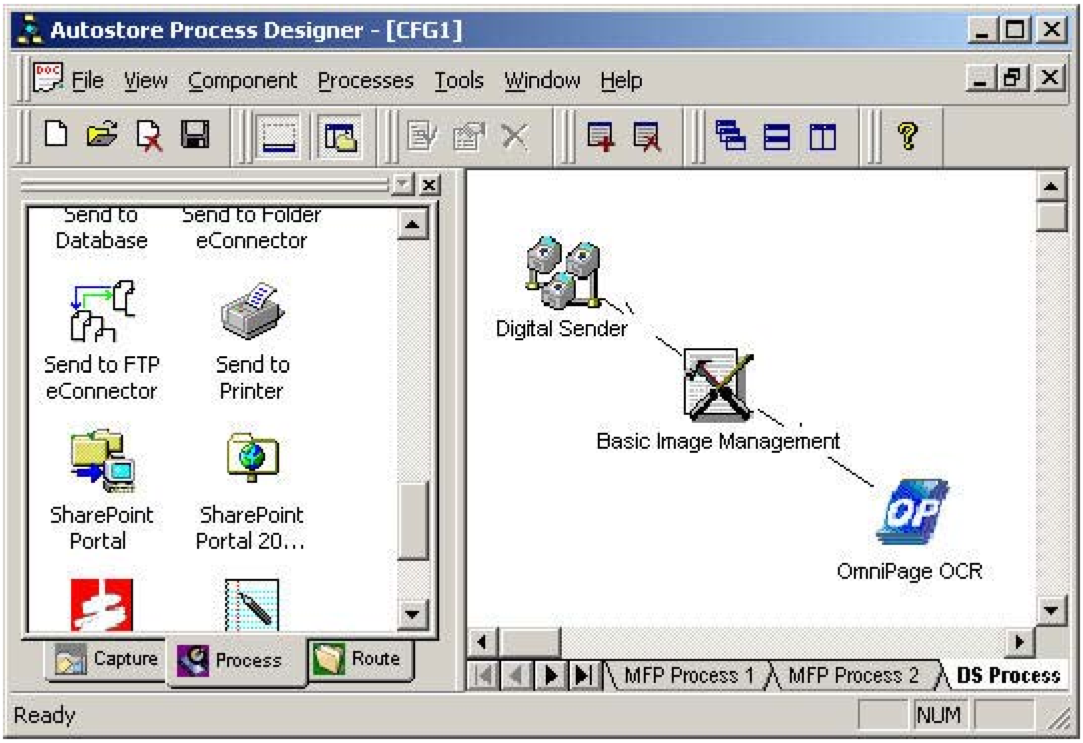
Frequency (H-M-S): 0 -- 0 -- 18

- Click the **Capture** tab under the component tray. Drag and drop the Digital Sender Capture component to the right-side pane.

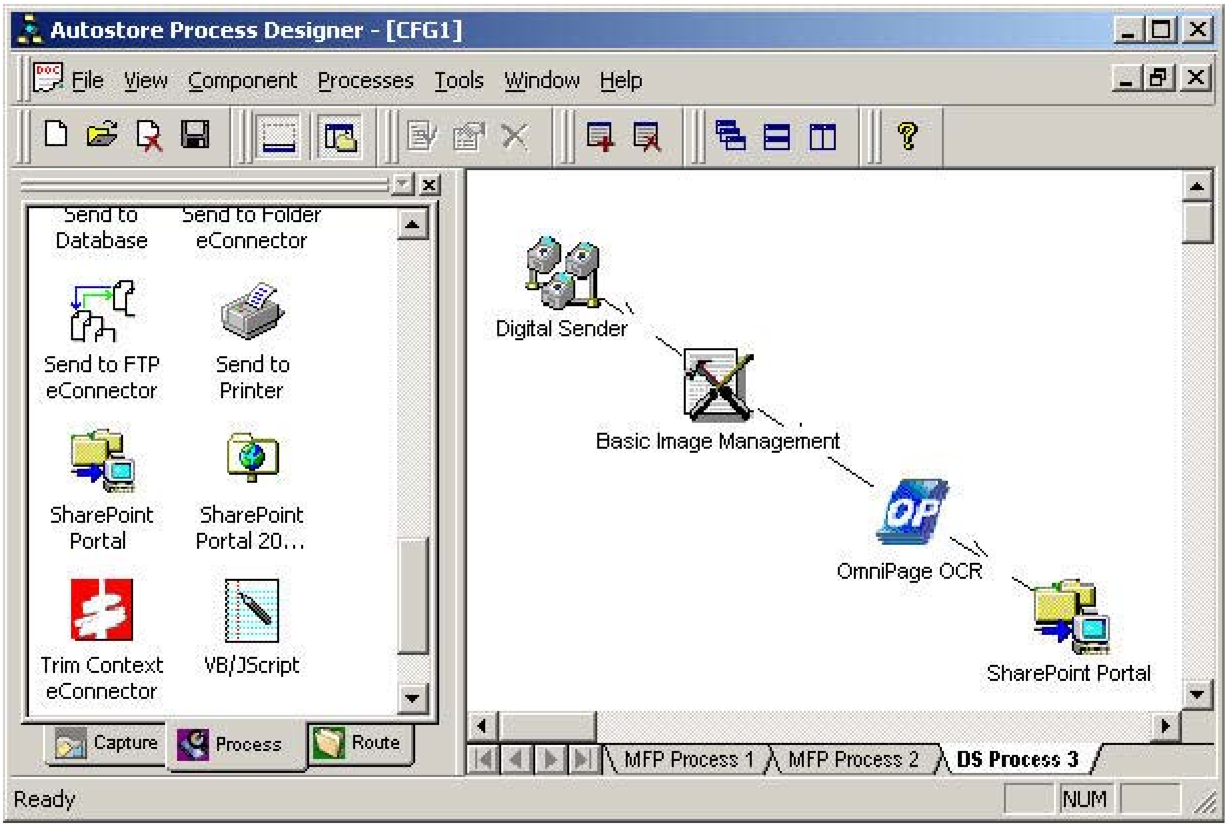


Autostore Process Designer - [CFG1]

- Click the **Process** tab under the component tray. Drag and drop the Basic Image Management Process component to the right pane. Next drag and drop the OmniPage OCR Process component to the right-side pane.



- Click the **Route** tab under the component tray. Drag and drop the SharePoint Portal Route component to the right-side pane.



This completes the third process in the configuration file. To save the configuration file, use the following procedure.

- On the Process Designer toolbar, click **File**, and then click **Save As**.
- Select the folder in which you want to save your file, type a name for the file, and then click **Save**.
- To exit the AutoStore software, go to **File**, and then click **Exit**.

You have created one configuration file that contains three processes. The three processes start simultaneously, but run at separate times because of the offset frequencies (18, 19, and 20) that you specified.

The first process used the template that is named MFP Scan OCR To Folder.CTF.

The second process used a blank process with one Capture component, one Process component, and one Route component.

The third process used a blank process with one Capture component, two Process components, and one Route component.

Starting the AutoStore service

AutoStore must be running as a service on a PC with the appropriate operating system before your process can work correctly.

1. Take one of the following actions to start AutoStore:

   On the computer desktop, click **Start**, select **Programs**, select **Hewlett-Packard**, select **HP AutoStore**, and then click **AutoStore Service Manager**.

   On the computer desktop, click **Start**, select **Settings**, select **Control Panel**, and then click **AutoStore Service Manager**.

   The **AutoStore** dialog box appears.

2. Type the field names from the following table, and then click **Apply**.

Field name	**Description**
Status	This is a display-only field that shows the status of the service. The default status of the service is **Stopped**.
AutoStore Script	This field contains the configuration file name. The AutoStore Process Designer generates this file when you save your AutoStore parameters. The file has a .CFG file extension. Either type the address or click the ellipses button to select a configuration file in the Windows Explorer view. A drop-down box shows recently-used scripts.
Startup Type	Using this field, you can select the startup type for the service. **Automatic** restarts the service automatically if the server is restarted. **Manual** requires the service to be restarted manually every time the server is restarted. **Disabled** marks the service as disabled so that it cannot be started.
Log On NT As	This field contains three additional fields, **Account**, **Password**, and **Confirm Password**. ● **Account.** Type the user name of the local computer where AutoStore is installed. The default is **LocalSystem**. If the folder where you store processed files is located on a remote computer on the network, you must provide proper credentials (at a minimum, provide write permissions) to the folder. Type the **Domain Name** followed by the **Account Name** (DOMAIN NAME\NT USER ID). ● **Password.** Specify a valid password. The default password for **LocalSystem** is blank. Do not type anything in the space provided. ● **Confirm Password.** Type the same password that you specified in the **Password** field. If you left it blank, then leave this field blank.

4 AutoStore processes

A process is a series of components that are connected in a certain order to correctly capture, process, and route information. A typical process consists of the following types of components:

- **Capture**. These components are responsible for capturing the input data stream in the process. Information can be moved from one Capture location (a source) to a Route location (a destination) by designing a two-component process. Add one Capture component to capture the information, and add one Route component to route and store the information in a software program or database. Your process must contain one Capture component.
- **Process**. These components are responsible for manipulating, extracting information from, converting, and formatting the data stream. A Process component makes data available to other components within the same process. Your process can contain no Process components or multiple Process components.
- **Route**. These components are responsible for connecting, routing, and storing information. When you use Route components with the enterprise connector capabilities, you can route information to one destination and link back to the information in another software program. Use this method of linking to route and link two destinations within one process. Your process must contain one Route component.

A process consists of at least two components: one Capture component and one Route component. When the components are added to a process and configured, the process is saved as a configuration file with the extension .CFG. Then, when you start the AutoStore Service Manager, you select the configuration file that contains the process that you want to run. More than one AutoStore process can run at a time. Each component within a process works with AutoStore Service Manager as a separate multi-threaded process.

NOTE You must configure the components before you can run your process in the Service Manager. If you have not already configured or updated the components, go back to the previous chapter and review Step 3.

Understanding AutoStore components

When you open the AutoStore Process Designer (click **Start**, select **Programs**, select **Hewlett-Packard**, select **HP AutoStore**, and then click **AutoStore Process Designer**), the available components appear in the component tray of the Process Designer window. To switch between Capture, Process, and Route components, click the tabs at the bottom of the component tray.

Components are the basic building blocks of processes. Each component can perform a defined task on data or images. Based on the process attributes, each component performs a designated task of reading, manipulating, or storing the data and files.

The following table lists the components that are installed with the AutoStore Server software, and additional components that work with AutoStore. For components that are not described in the *AutoStore Support Guide*, see the Help file that is associated with that component.

AutoStore components

AutoStore component type	Components that are included with AutoStore Server software	Other components that are available for use with AutoStore
Capture components	● HP 9100C Digital Sender ● Knowledge Package Loader ● MFP 4100/9000 ● HP LaserJet 9055mfp or HP LaserJet 9065mfp ● Poll Directory	● POP3 Email ● Batch Importer ● ABM Exporter ● AutoCapture Server

AutoStore components (continued)

AutoStore component type	Components that are included with AutoStore Server software	Other components that are available for use with AutoStore
Process components	● Send to Printer ● Knowledge Package Loader ● Knowledge Package Builder ● ABBYY Fine Reader OCR version 6.0 ● SharePoint Portal 2003 ● Omni Page OCR ● Basic Image Management	● VB/JScript ● Watermark ● Notification ● ABBYY FormReader 6.0 ● PDF 417 Barcode ● Professional Barcode ● Professional Image Management ● File Options ● Data Interchange ● Send to Database ● SharePoint Portal v1.0 eConnector ● ApplicationXtender ● Hummingbird eConnector ● Send to Folder eConnector ● Send to FTP eConnector
Route components	● LAN Fax ● Folder Store ● FTP Store ● Send to Printer ● Send to Mail Recipient ● Send to PC ● Send to Database ● SharePoint Portal 2003 ● Multi Router ● Lotus Notes/Domino ● Domino.Doc ● Microsoft Exchange	● Documentum ● Send to Database (Bulk) ● iManage ● VB/JScript ● Hummingbird ● FileNET v3.1 ● ABM Importer ● IXOS ● SharePoint Portal v1.0 ● ApplicationXtender ● Send to Folder ● Send to FTP

Types of components

Use the following types of components to build an AutoStore process.

- **Capture components.** All AutoStore processes must start with one Capture component. A Capture component reads data, images, or other types of files within an AutoStore process.
- **Process components.** This type of component retrieves data, extracts information, converts formats, or manipulates the content of information. The Process components are important for processing the data content of files.
- **Route components.** All AutoStore processes must end with one Route component. This type of component forwards the information to its final destination. The destination could be an e-mail address, a database file, or an FTP site.

NOTE A Capture or Process component can be further categorized as a Mapping component based on your configuration requirements. A Mapping component maps the processing attributes of other components to its own internal properties. For example, the Digital Sender component maps the attributes for OCR, form recognition, SharePoint Portal Server, and so on to each digital sender, function key, and application function key.

Understanding AutoStore processes

Use the Process Designer to assemble, configure, and save a process in a configuration file. A process must contain one Capture component and one Route component. A process can contain any number of Process components.

An AutoStore configuration file can contain one or more of the following types of processes:

- **Autonomous.** These are independent processes that do not feed information to each other.
- **Multiprocess chain.** These are connected processes that feed information to each other through files or other means.

Designing an AutoStore process

You can use a process to connect, process, and route information from any Capture component to any Route component. The first step in creating your process is to design an AutoStore process separately (perhaps write down your idea on a piece of paper), and then use the AutoStore Process Designer to create a process.

The process-design procedure is fairly simple and can be broken down into a number of steps. Follow the design steps that are described in the following sections to design an AutoStore process. If your process is a simple single-thread process (for example, read from one or more digital senders and store data in the SharePoint Portal), you do not need to design your process separately. You can simply create the process in the AutoStore Process Designer and activate it in the AutoStore Service Manager. The following design guidelines are meant to help you create more complex, multi-threaded AutoStore processes.

Step 1: Define the process attributes

The first step in designing a process is to create an outline that contains the following information:

- The type and format of the information that you will be processing (such as .TXT, .TIF, CAD [computer-aided design], .PDF, or .BMP)
- The amount of information that you will be processing
- The frequency of the input data stream
- The media for this information (such as directory or SMTP e-mail)

Use this information to estimate the number of AutoStore servers that are required, and the configuration hardware for each. The estimate should be based on sample performance data that can be calculated by running the applicable sample data through the AutoStore server. The benchmark results vary depending on the server configuration, data size, and processing steps.

Step 2: Create a list of required processes

Based on the following process-design parameters, decide how many separate processes you need:

- A process can contain only one Capture component. For example, the process can use either the Poll Directory or Digital Sender component, but not both.
- A process can contain only one Route component. For example, the process can use FTP Store or Microsoft Exchange, but not both.
- A process can contain any number of Process components. The order in which the components are completed within a process is static. For example, to create the same two Process components in two different orders, you need to create two separate AutoStore processes.
- One or more processes can run simultaneously on the same AutoStore server.
- AutoStore is a multi-threaded program that can run many processes at the same time.

NOTE To have the results of one process feed into another process, create two processes and chain them together.

Examples of processes

- **Situation:** You have 20 digital senders with programmed soft keys that send .HPS files and images to one Inbox directory on the server. The images must be split into individual documents every two pages, converted to Microsoft Word documents by using OCR, and stored on a SharePoint Portal workspace.
  - **Number of processes needed:** One
  - **Reason:** All digital senders feed to one inbox (one input Capture component), and all Word documents are stored in the SharePoint Portal Server (one Route type).
- **Situation:** You want 20 digital senders with programmed soft keys to send .HPS files and images to two separate directories (two input Capture components).
  - **Number of processes needed:** Either One or Two
  - **Reason for one process:** You have two inbox directories. You could use the File Options Process component before the Route component. This allows you to send files to a single destination, and then on to the Route component.
  - **Reason for two processes:** You have two inbox directories. Each Input directory can have a separate process. Note that this is *not* a multiprocess chain, in which one process feeds into the other process.
- **Situation:** You want 20 digital senders with programmed soft keys to send .HPS files and images to one Inbox directory. The images must be split into individual documents on every two pages, processed by using OCR to convert them to searchable .PDF documents, and stored on a SharePoint Portal workspace. Each image also must be converted to an .HTML file format and stored in a directory.
  - **Number of processes needed:** Two
  - **Reason:** You need one process that contains a SharePoint Portal Route component and one that contains a Folder Store Route component.

Step 3: Create a list of components for each process

Capture component requirements: Create a list of your Capture components by categorizing your input capture types. If the source of the information is a digital sender that generates an .HPS/image file pair, then you must use a Digital Sender Capture component. If the source is another file type, then use the appropriate Capture component.

Process component requirements: Based on your program requirements, you must decide which Process components are suitable for your needs. If you have custom requirements, you can contact HP support for information about creating custom-built components that match your information technology (IT) processing requirements.

The Process component features are in three major categories:

- **Conversion:** These Process components convert the data-stream format. For example, the OCR Process component converts the image files to various types of text files. Use conversion when you need to modify the data format, presentation, or searchability.
- **Connectors:** Use Process components with connector capability to store files in one program and place a link (for example, a URL link pointing back to the location of the document) in another program field. Using this feature, you can place files in your document-management system and link back to them from other software programs.
- **Extractors:** Barcode and text parsers are extractor-type Process components. These Process components extract information from images and make them available for use.

Decide which types of components you want to use and the order in which you want to use them. Remember that the order of components is important. For example, you can split images into separate documents every two pages and use OCR to convert them to .PDF files, or use OCR to convert all pages to one .PDF file and then split that image into multiple documents. In the first case, the AutoStore process produces multiple two-page images and corresponding two-page .PDF files. In the second case, the AutoStore process generates a single .PDF file that contains multiple two-page images. To obtain the result that you want, you must arrange components in the correct processing order.

Route component requirements: Create a list of the different Route components that your process requires. For example, if you need to have files processed and stored in SharePoint Portal Workspace, a FTP site, and Microsoft Exchange, then include these components in your list.

Step 4: Decide on your overall process chain

For each process, create a chain of components based on your software program requirements. Consider the following parameters when creating your Process component chain:

- Each component operates only on its defined input type (for example, the OCR Process component does not operate on text files) and passes through all other input types.
- Components are completed from Capture component to Route component.
- Components that depend on output from other components must be used after those components.

Step 5: Decide if you need a multiprocess chain

Consider creating a multiprocess chain when the processed output from one process must be stored in multiple locations. For an example, if your process requires that images be converted to .PDF files and stored in a folder, and that the files also be sent as e-mail attachments, then consider chaining two processes together. The first AutoStore process converts the images to a .PDF file with a Folder Store Route component. The second process reads the .PDF files from the destination folders and sends them to an e-mail address as attachments.

For examples of how to use multiprocess chains, see the multiprocess chain section of this document or the specific Help file that is associated with each Capture component.

Step 6: Setting attribute mapping for each process

If you are designing a process that contains a digital sender Capture component, you need to define the process attributes for some of the soft keys by defining Routing Slips. In the AutoStore Process Designer, double-click the Digital Sender component to define your routing slips. If your process contains an MFP, you can also create attributes (form fields), and if your process contains the Knowledge Package Loader component, you can map attributes in .XML file formats to fields in that component.

Creating AutoStore processes

Use the following instructions to create AutoStore processes and save each process in a configuration file.

Tips for creating AutoStore processes

Use the following tips when creating a process by using the AutoStore Process Designer.

- A process must have a Capture component and a Route component.
- A process cannot have two Process components of the same type.
- If a Mapping component is used within a process, then all attributes for the components that follow it must be configured within the Mapping component. (A Mapping component maps the processing attributes of other components to its own internal properties. For example, the Digital Sender component maps the attributes for OCR, form recognition, SharePoint Portal Server, and so on to each digital sender, function key, and application function key.
- Set the processing timer to a reasonable value for your process. The processes that have extremely short frequency timers might deplete the hardware resources.
- The order of components within a process is important. For example, a process that has OCR and image-processing components (in that order) uses OCR with the images first and then completes the image processing (by splitting the document). In this scenario, the output of the process is a single OCR text document, and then multiple split images. In another scenario, a process that uses the image processing (split-document) component followed by OCR would create multiple split images and then create multiple OCR text documents.

To create a new AutoStore process

1. Open the AutoStore Process Designer. Click **Start**, select **Programs**, select **Hewlett-Packard**, select **HP AutoStore**, and then click **AutoStore Process Designer**.
2. On the Process Designer toolbar, click **File**, and then click **New**.
3. Select **Blank Process** in the **New** dialog box, and then click **OK**.
4. The **Process Information** dialog box appears. This dialog box contains the process attributes. Type a name for the new process in the **Process Name** field.
5. In the **Frequency** fields, type the frequency at which you want the process to run. The frequency is specified in hours, minutes, and seconds and represents how often the Capture component communicates with the AutoStore Service Manager. The longer the frequency, the more time the process takes to run.
6. Click **More**. The following fields become available.
   - The **Debug** check box is used only for support purposes. Only the AutoStore administrator should turn on this feature. When Debug is activated, the various components within AutoStore start creating log messages within the AutoStore Log file. These messages are designed to help the support staff take a closer look at the internal operations of an AutoStore server that is running in the background.
   - In the **Threshold** field, use the spin box to specify a number between 0 and 10, with 0 representing the lowest level of logging and 10 representing the highest level of logging. The default is set to 4. When the Debug function is activated, the number of log messages from each component increases based on the Threshold level that you select. This operation requires free disk space and can affect system operations if it is not turned off. System performance and availability might be affected if the hard-disk storage is depleted.
7. Click **OK**.
8. Drag and drop a Capture component from the component tray to the blank process in the right-side pane.
9. Click the **Process** tab under the component tray to see the available Process components. Skip to step 11 if you are not adding any Process components.
10. Drag and drop one or more **Process** components from the component tray to the process in the right-side pane.
11. Click the **Route** tab under the component tray to see the available Route components.
12. Drag and drop one Route component from the component tray to the end of the process in the right-side pane.
13. On the AutoStore Process Designer toolbar, click the **Save** button.
14. In the **Save As** dialog box, select the folder in which you want to save your file, type a name for the process, and then click **Save**.

NOTE You can reposition components by dragging them around on the process design screen. The screen location of a component on the process design screen is not related to its operation. To change the order of components within the same process, you must remove a component and then replace it at the new location within the process.

To create a new process by using an AutoStore template

A process template is a special kind of configuration file that provides basic tools for shaping an AutoStore process. Templates can contain one or more processes, process attributes and settings, and component attributes and settings. AutoStore provides two templates, which are installed with the AutoStore software. You can also create your own templates for storing components or settings that you want to reuse in other processes.

1. In the AutoStore Process Designer, click **File**, and then click **New** or right click in the right-side pane and click **New Process**.
2. Select the template that you want from the list of templates and click **OK**. Click the **Browse** button to locate other template files.
3. Position the cursor in the right-side pane, away from any of the Process components. Double-click to open the process attributes.
4. Specify the process attributes or modify any existing attributes as needed, and then click **OK**.
5. On the AutoStore Process Designer toolbar, click the **Save** button.
6. In the **Save As** dialog box, select the folder in which you want to save your file, type a name for the process, and then click **Save**.

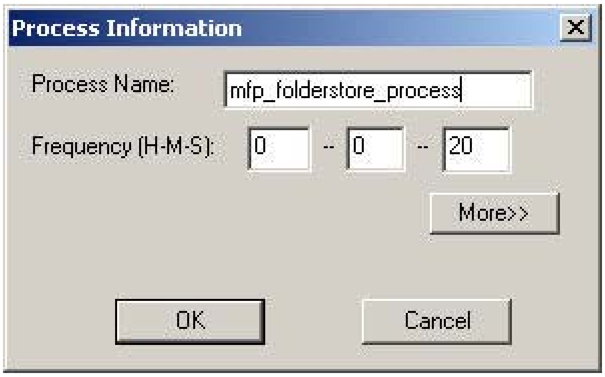
Creating and testing a simple AutoStore process

Use the following example to start creating and testing AutoStore processes.

Example: Creating a new process that uses a Capture and Route component

The following procedure is an example of how you can create a new process that uses the MFP (4100/9000) Capture component and the Folder Store Route component.

1. In the AutoStore Process Designer, on the toolbar click **File**, and then click **New**. In the **New** dialog box, click **Blank Process**, and then click **OK**.
2. In the **Process Information** dialog box, type a name for the process (for example, `mfp_folderstore_process`), and then click **OK**.

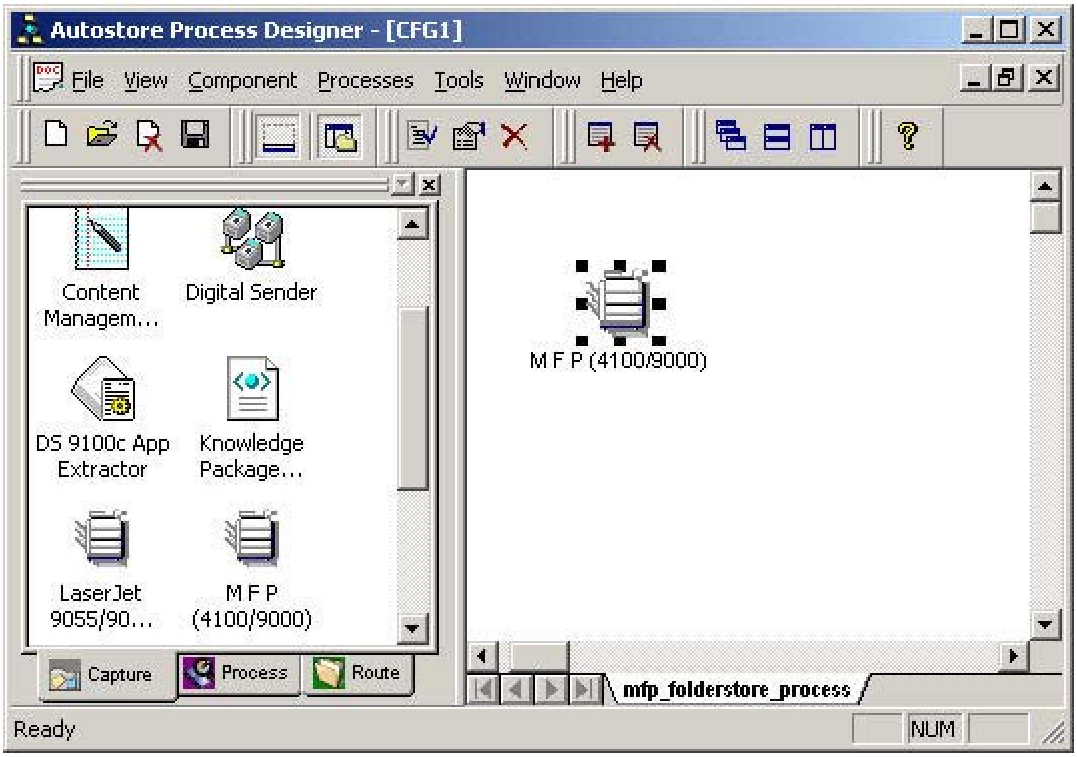


Process Information

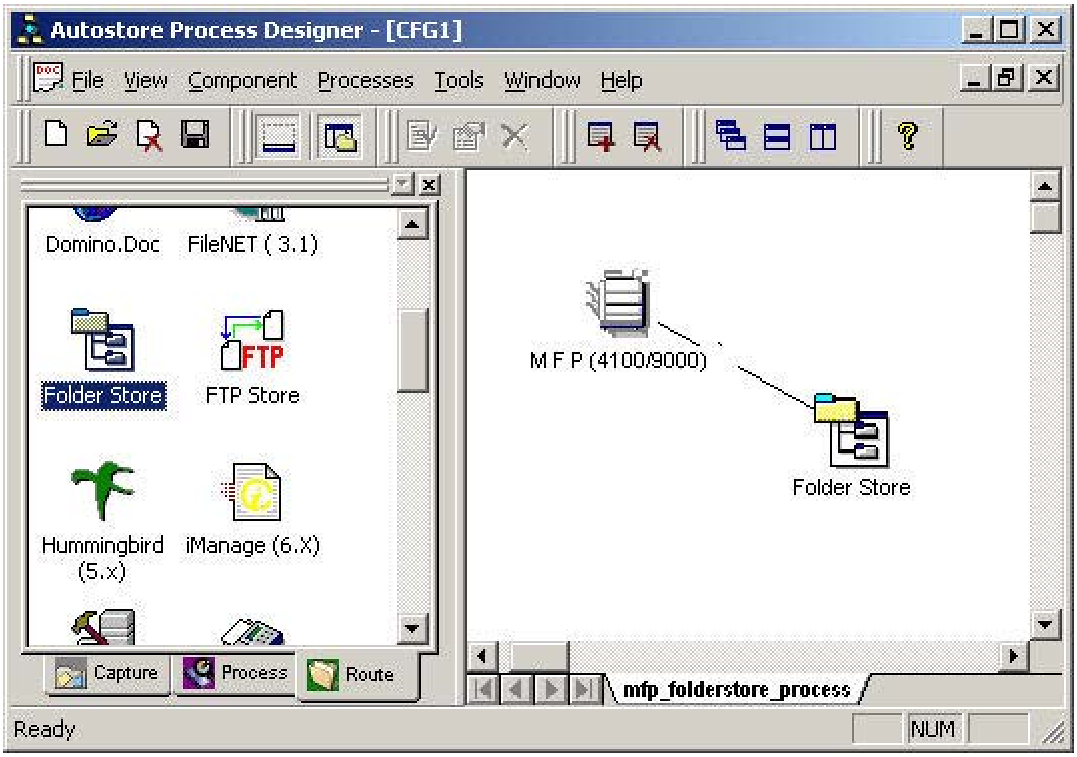
Process Name: mfp_folderstore_process

Frequency (H-M-S): 0 -- 0 -- 20

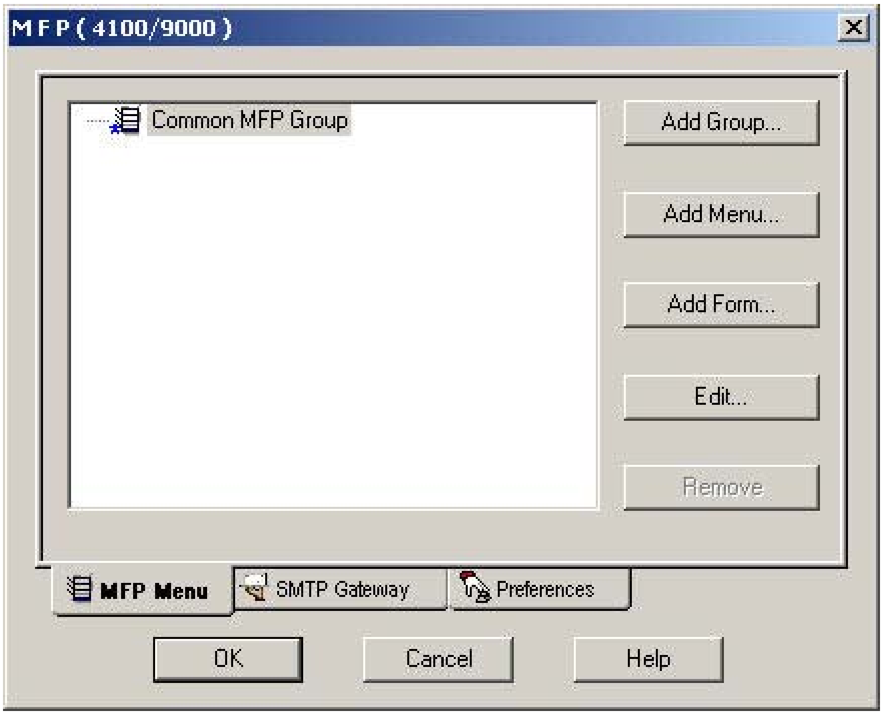
3. Click the **Capture** tab and drag the MFP (4100/9000) component into the right-side pane.



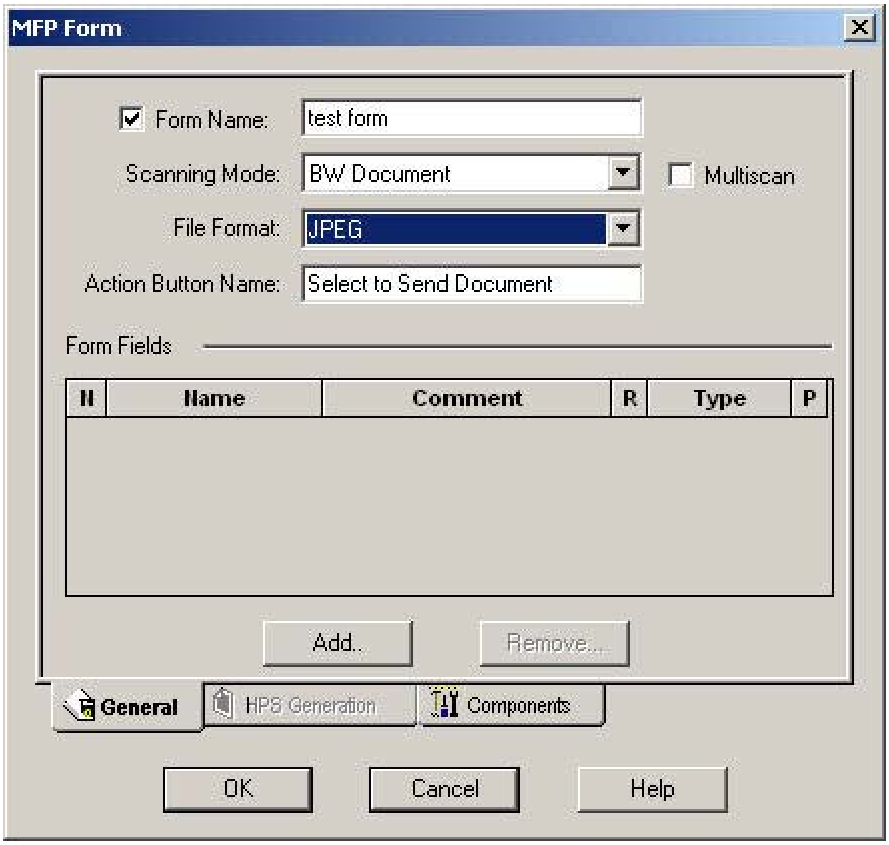
4. Click the **Route** tab and drag the Folder Store component into the right-side pane.



5. Double-click the MFP (4100/9000) component to gain access to the configuration dialog box. The **Common MFP Group** appears. Unless you create another group, all MFPs are contained in the Common MFP Group and inherit the menus that have been created for this group.



6. Click **Add Form**. On the **General** tab, type a name for the form (for example, `test form`). Select the scanning mode and file format, and then type the action-button name (for example, `Select to Send Document`).



MFP Form

☑ Form Name: test form

Scanning Mode: BW Document □ Multiscan

File Format: JPEG

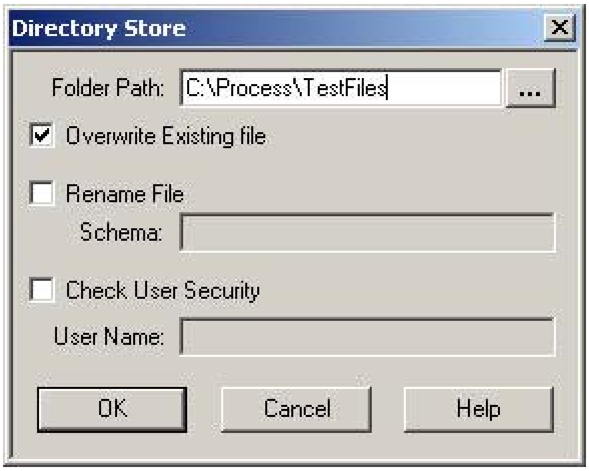
Action Button Name: Select to Send Document

Form Fields

N	Name	Comment	R	Type	P

7. Click the **Components** tab, and then click "**...**" to browse for a path name.

8. Select the folder path and select the **Overwrite Existing File** check box.



Directory Store

Folder Path: C:\Process\TestFiles

☑ Overwrite Existing file

□ Rename File

Schema:

□ Check User Security

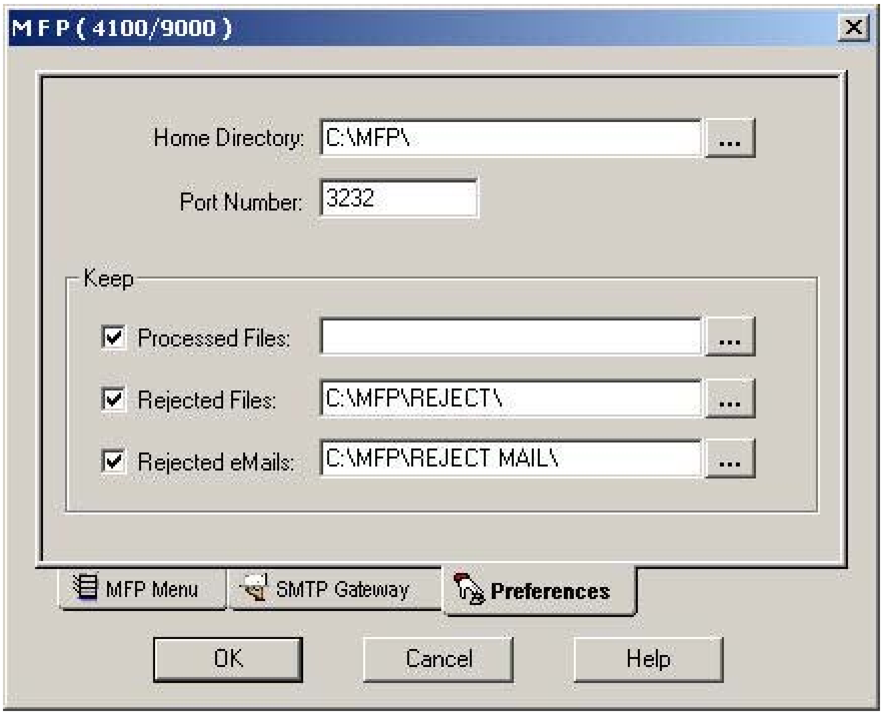
User Name:

9. Click **OK** to close the form.

10. Click the **SMTP Gateway** tab and type the hostname or IP address of at least one SMTP gateway. Note that because AutoStore uses the SMTP protocol to turn on the MFP component, this gateway is used to route e-mails that are sent from the MFP to the e-mail destinations.

NOTE The AutoStore Service Manager must be started before the MFP e-mail function turns on. When the Service Manager stops, the MFP e-mail functionality stops.

11. Click the **Preferences** tab and type information for the following working directories: Home Directory, Processed Files, Rejected Files, and Rejected Emails. The port should be set to **3232**.



MFP (4100/9000)

Home Directory: C:\MFP\

Port Number: 3232

Keep

☑ Processed Files:

☑ Rejected Files: C:\MFP\REJECT\

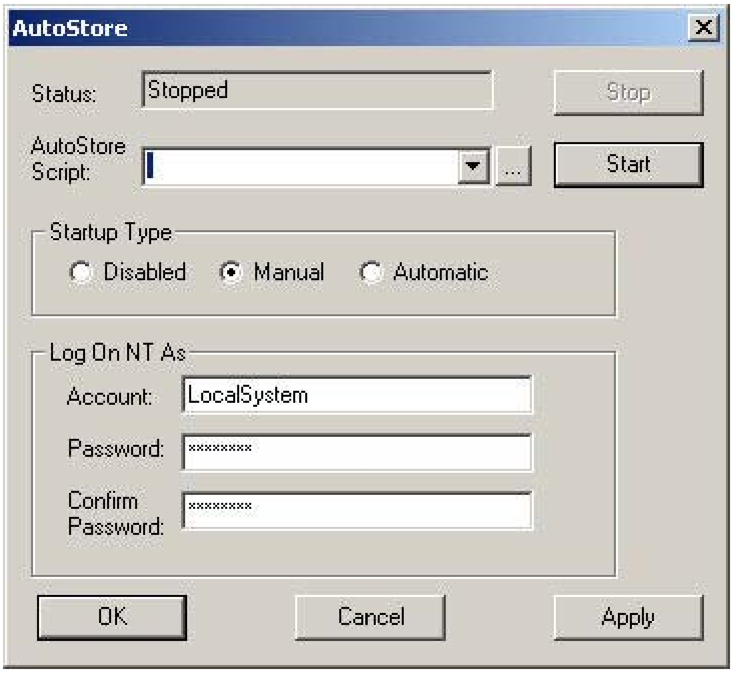
☑ Rejected eMails: C:\MFP\REJECT MAIL\

12. Click **OK** to close the MFP component configuration.

Example: Configuring the AutoStore Service Manager to use a process

After you have created and saved an AutoStore process, you must configure the AutoStore Service Manager to use the process, and then start the AutoStore Service Manager.

1. On the Windows taskbar, click **Start**, select **Programs**, select **Hewlett-Packard**, select **HP AutoStore**, and then click **Autostore Service Manager**. The **AutoStore Service Manager** configuration dialog box appears.



AutoStore

Status: Stopped

AutoStore Script:

Startup Type: ○ Disabled ● Manual ○ Automatic

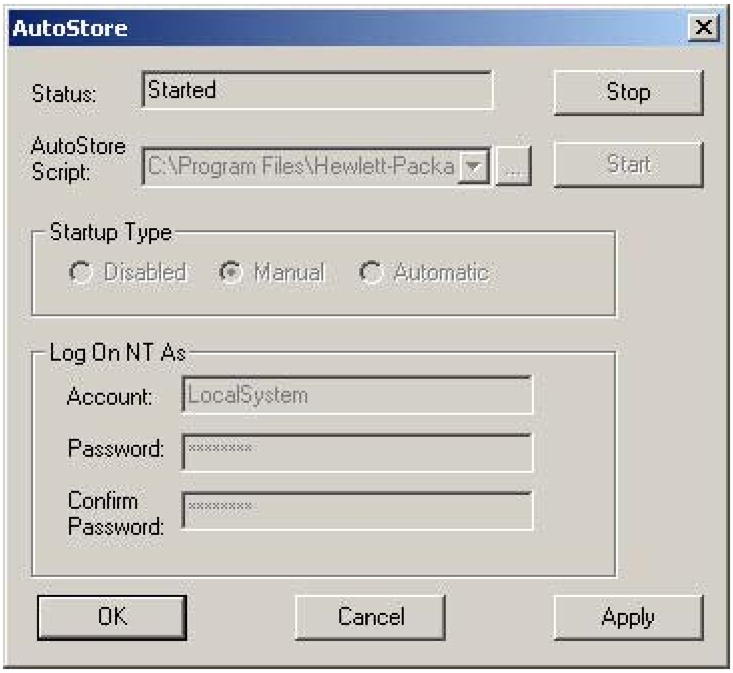
Log On NT As

Account: LocalSystem

Password: ********

Confirm Password: ********

2. Click "**...**" and browse for the .CFG file that you created and saved in the AutoStore Process Designer (for example, `mfp_folderstore_process`).



AutoStore

Status: Started

AutoStore Script: C:\Program Files\Hewlett-Packa

Startup Type: ○ Disabled ● Manual ○ Automatic

Log On NT As

Account: LocalSystem

Password: ********

Confirm Password: ********

3. Click **Apply**, and then click **Start**. For this example, leave the Startup Type for the AutoStore Service Manager set to **Manual**.

Testing an AutoStore process

When you create a new AutoStore process, you might want to test the process before you make it available to all users. To test the AutoStore process that you created, place a document on the flatbed scanner or in the automatic document feeder (ADF). On the device control panel, press **Menu**, select **Send To**, and then select **Send Document.** (This is the **Action** button string that you typed in the AutoStore Process Designer in the **Form** dialog box.) Depending on the size of the document and the file format that you selected, the file appears within a few minutes in the destination folder that you defined in the Folder Store configuration.

Modifying AutoStore process attributes

By using the AutoStore Process Designer, you can view and modify the attributes of an existing process. Complete the following instructions to modify process attributes.

To modify the attributes for a process

1. In the AutoStore Process Designer, click **File**, and then click **Open**.
2. In the **Open** dialog box, browse to the configuration file that contains the process that you want to modify. Click **Open**.
3. Position the cursor in the right-side pane, away from any of the Process components. Double-click to open the process attributes.
4. Modify the process attributes that you want to change. Use the following table to modify attributes.

Field name	**Field description**
Process Name	The name for the process. The process name becomes the task name within the multi-thread AutoStore processor.
Frequency	The frequency of completion for this task. The completion is specified in the following units: ● H – hours ● M – minutes ● S – seconds
Debug	This item is used only for support purposes and should be activated only when requested by the AutoStore Administrator. When the Debug feature is activated, the various components within AutoStore start creating log messages within the AutoStore Log file. These messages are designed to help the support staff take a closer look at the internal operations of an AutoStore server that is running in the background.

Field name	Field description
Threshold	This field has a valid range of 0 to 10, with 0 representing the lowest level of logging and 10 representing the highest level of logging. **NOTE** When the Debug feature is activated, the number of log messages from each component increases based on the Threshold level that you selected. This operation requires free disk space and can affect system operations if it is not turned off. System performance and availability might be affected if the hard-disk storage is depleted.

Deleting an AutoStore process

1. In the AutoStore Process Designer, click **File**, and then click **Open**.
2. In the **Open** dialog box, browse to the configuration file that contains the process that you want to delete. Click **Open**.
3. On the Process Designer toolbar, click **Processes**, and click **Delete Process**. You can also right-click in the right-side pane of the Process Designer and click **Delete Process**.
4. Click **Yes** to confirm. The configuration file and the process that it contains are deleted.

Creating a multiprocess chain

A multiprocess chain is a set of processes in which output from one process feeds into the input of another process. Process chains are useful when you have images or data elements that must be routed to multiple destinations. For example, a process that requires routing to SharePoint Portal Server, Microsoft Exchange, and SMTP e-mail must be designed as a three-process chain. The first process routes information to the SharePoint Portal Server and, on success, the process stores the files in a folder destination for use in a second process. The second process uses the Poll Directory Capture component to grab the files from the directory and route them to Microsoft Exchange. The third and final process uses the SMTP Route component to route the same files by e-mail.

Process chains require the output from one process to feed into the second process. Use the following tips when creating process chains:

- Two methods are available for creating a process chain.
  - Make the **Success** or **Failure** directory of one process feed into the inbox of the next process, and use another process to read the images. For example, the first process has a Digital Sender Capture component that includes a **Success** directory of C:\PROCESS1\SUCCESSDIR. The next process in the chain uses a Poll Directory Capture component and sets its Inbox directory to C:\PROCESS1 \SUCCESSDIR. Note that with this scenario, the files placed in Process 2 are the same files that were placed in Process 1. (In other words, the same files were fed to both processes in this chain.)

    Use this method if the same files must be routed to different destinations. Note that the second process in this example uses the Poll Directory Process component and therefore does not read the .HPS file. With this example, the corresponding data elements that are associated with .HPS file are not carried over to the second process in the chain. Use the Digital Sender Capture component in the second process to carry the .HPS file into your process chain. When using the Digital Sender Capture component in the second process, you must set the file type to .TIF or .PDF to process the .HPS file correctly.
  - Use the File Options Process component to store a copy of a particular file type that a process generates into the Inbox directory of another process. This technique is more effective when you require the processed files to be routed to multiple destinations. For example, Process 1 reads .TIF file images and converts them to searchable .PDF files (by using the Professional OCR Process component). Before you store the files in the SharePoint Portal Server, a file options filter is used to store the .PDF files in a directory. The second process uses the output of the File Options Process component as its inbox, reads the files, and sends the searchable .PDF files to Microsoft Exchange.
- When using the Digital Sender Capture component in a process chain (reading files from the Success or Failure directory of another process) to read HPS/Image file pairs, you must set the file type to .TIF or .PDF.
- Use appropriate frequencies for each process. Do not use high-frequency polling for your processes. A process that uses a high frequency rate could deplete the hardware processing resources.
- When using process chaining, consider various scenarios of failure and success in routing each segment of the process. Design each segment of your process by giving consideration to failure as well as success in routing each message.
- The failure of a process chain can be fed into a “failure notification” process that has an Inbox directory that is designated for catching failed routings from all processes.

5 Capture components

AutoStore uses a Capture component to create a work object for a process. The Capture component must always appear at the beginning of a process. The Capture component initiates a process by capturing data (data files from the Inbox directory, data elements from the database file, and so forth) and delivering it to the Process or Route components. Each process must have a single Capture component.

Add Capture components at the beginning of each process. A Capture component can be a mapping component. You can replace the Capture component in a process by dragging a replacement Capture component from the component tray of the Process Designer to the right pane (where the current process appears). The Process Designer asks you to confirm that you want to replace the Capture component. After you confirm, the new Capture component appears in the process.

Digital Sender component

The Digital Sender component provides three major functions:

- Reads and processes .HPS files
- Decrypts .HPS and image files
- Maps digital-sender soft keys to the Process-component attributes

The Digital Sender component is a mapping component and can map all of the other component attributes to function keys.

When you are designing a process that requires working with digital sender image files, your process must start with the Digital Sender component.

Design your process by first adding the Digital Sender component as your Capture component. Then add the appropriate Process and Route components.

When your process design is completed, you can start mapping the processing attributes to soft keys by defining routing slips. A routing slip refers to the component attributes defined for configuring a function key on a digital sender (or a form on an MFP). The routing slips are defined within the Digital Sender or MFP component attributes.

By adding this Capture component to your process, you can accept the following types of input:

- **.HPS file format.** An HP-designed file format that carries the interface parameters for digital senders, this file contains, in text format, all of the index information that users enter at the front panel of the digital sender.
- **.TIF file format.** Use the .TIF file option only if your .TIF files are going to arrive at the inbox later than the .HPS files (usually this occurs when a program processes the .TIF files before they are placed in the Inbox directory) and you want your process to wait for the .TIF file and match it to the .HPS file before starting the process. This is called matching .TIF. This option also updates the file path name in the .HPS file to match the image file.
- **.PDF file format.** Use the .PDF file option only if your .PDF files are going to arrive at the inbox later than the .HPS files (usually this occurs when a program processes the .PDF files before they are placed in the Inbox directory) and you want your process to wait for the .PDF file and match it to the .HPS file before starting the process. This is called matching .PDF. This option also updates the file path name in the .HPS file to match the image file.

The following describes .PDF and .TIF matching: This is used only when a separate application is processing the .HPS + .TIFF file (or .HPS + .PDF file) prior to these files arriving into the AutoStore inbox. The application that is processing these files could store them into the AutoStore inbox in any order (for example, the .HPS file arrives first, then the .TIF file arrives later). Therefore, AutoStore cannot start processing the .HPS file without ensuring that the .TIF file is also there. By using this matching option, AutoStore can ensure that the file pair (.HPS + .TIF or .HPS + .PDF) exists before processing the workflow. Futhermore, when an .HPS + .TIF or .HPS + .PDF file pair is moved out of the Digital Sender Service inbox, the .HPS image folder path is no longer valid (pointing to the old location). By using this feature, the .HPS image folder path is updated by AutoStore to correspond with the matching image within the same directory. Note that this is not only helpful but necessary. Without updating the image folder path within the .HPS file, the .HPS image folder path is pointing to an incorrect location.

Feature highlights

You can use the Digital Sender component to read and process data and image files that are generated by using the digital sender function key. The Digital Sender component is designed to handle all key data types and images, as well as the decryption of secured files.

The Digital Sender component is also used to create routing slips for soft keys. With this feature, you can build a custom routing slip for each soft key and its connection to your software program.

Configuring the Digital Sender component

Use static or dynamic values, as defined in the Capture component Runtime Replacement Tags (RRTs), to set the case-sensitive attributes for the Digital Sender component.

The following attributes are available in the **Digital Sender** dialog box.

Workspace tab

Use the workspace parameters to specify the Inbox and Working directories, as well as the order in which the files are introduced into the process.

NOTE The Digital Sender Link software must be running on the same machine that the server is running so that the Digital Sender component can communicate with the server. You also must ensure that the Inbox directory is available. The Digital Sender component does not create an Inbox directory by default. However, the Working directory is created by default.

The following table describes the fields that are available on the **Workspace** tab.

Field name	Description
Input Directory	The directory that the Digital Sender component searches to find a file. Type the Directory name only (for example, `c:\SampleInboxDir\`).
Working Directory	The name of the directory that files move to after the process starts. Type the directory name only (for example, `c:\SampleWorkDir\`).

Field name	Description
File Type	By default, this field is set to HPS, but can accept any of the following parameters: **HPS.** Use this when the HP Digital Sender service generates .HPS files. The component will look for the .HPS file first and the corresponding image next. **TIFF.** Use this value when .TIF files will be processed by third-party programs and .HPS files are probably going to arrive before the .TIF files are processed and ready. This selection then looks for .TIF file images, finds the matching .HPS file, and updates the .HPS file to match the .TIF file path name. **PDF.** Use this value the same way as TIFF, except for .PDF files. When you expect the .PDF files to arrive late to the Inbox directory, use PDF so that the component looks for .PDF files, matches them with .HPS files (which arrived in advance of the .PDF files), and updates the .HPS file to match the .PDF file path name.
HPS Passthrough	**Pass-through.** When you select this check box, the .HPS file is copied to the destination location. Use this option to store processed images and .HPS files to the destination location.
On Success - Remove Files	After the process successfully completes the destination component, the .HPS file and the image file are deleted from the Working directory.
On Success - Move Files	After the process successfully completes the destination component, files are moved to the designated directory.
On Failure - Remove Files	If the destination component reports an error in completing its operation, remove the files from the Working directory.
On Failure - Move Files	If the destination component reports an error in completing its operation, files are moved to the designated directory.

Decryption tab

Decryption only applies if the files that are received to the inbox have been encrypted by the source that generated them. The encryption must be one of the supported standard encryption methods that are listed under the Algorithm field. The most common use of this encryption technology is when AutoStore is used both for encryption and decryption. The following table shows a typical use of the AutoStore secured transmission module.

NOTE If the decryption is activated, then all of the files in the inbox are expected to be encrypted. The encrypted files must include the .CRY extension (for Cryptography). Make sure that all of the files that are stored in the Inbox directory are in .CRY file format and include the correct encryption algorithm.

Field name	Description
Activate	Decryption activation starts the decryption processing for all received files. Note that encrypted files are expected to be in the .CRY file format and are only processed if the correct encryption algorithm and password are provided.
Algorithm	The supported encryption algorithm. Note that all files in the inbox must match your choice of encryption. Each inbox can support one method of encryption.
Key	Decryption key. This key must match the encryption key.
Key File	The file path name for the file that is to be used as key. The same file must be used to encrypt the files.

Digital Senders tab

The **Digital Senders** tab contains information about routing slips, soft function keys, and prompts, and you can also use the attributes on this tab to configure other components in the process.

Routing slip configuration

Use the **Digital Senders** tab to create a routing slip for each function key. Make sure that you understand the role of routing-slip entries. Each entry represents the processing attributes that the AutoStore process uses when it receives images from the entry. A routing slip refers to the component attributes defined for configuring a function key on a digital sender (or a form on an MFP). The routing slips are defined within the Digital Sender or MFP component attributes. Each routing slip contains the following information:

- The Process component attributes, such as OCR or image-processing component attributes
- The Route component attributes, such as which database or which form will be used

You can define a general routing slip first, and then define any number of specific routing slips. The general routing slip is defined for use by all digital sender entities (such as digital senders and function keys). A specific routing slip is defined specifically for an entity, such as a named digital sender, function key, or soft function key.

When AutoStore receives an image, the server first looks for a specific digital sender routing slip. If it does not find one, then AutoStore applies the general routing slip.

The same concept applies at all levels. For example, when processing an image, if AutoStore finds a specific function key routing slip, it uses that routing slip to process the image. If AutoStore does not find one, it looks for a specific function key routing slip. AutoStore applies the specific function key routing slip if it finds one. Otherwise, it looks for a specific digital sender routing slip, and uses that. If AutoStore does not find a specific digital sender routing slip, it uses the routing slip definition in the general digital sender routing slip.

By using this method to find the appropriate routing slip for images, AutoStore makes it easier to configure any required routing slips. In other words, you do not need to write a specific routing slip for each digital sender in your network, unless each digital sender has its own specific processing parameters. To illustrate this method, the following example describes a digital sender network:

- The network has 30 digital senders.
- Ten function keys are defined across all digital senders.
- Each function key has three destinations defined.

Case 1: You want all images from all digital senders to be placed in the same form on a single Exchange or Notes server, no matter what key you press on the digital sender. This design can be used for creating a common queue of images. Your queue-processing program then determines how images are processed after they are deposited in this database.

Case 1 solutions: Define a single-process, add a digital sender component and all other appropriate components, and then define the general routing slip. You do not need to define any additional specific digital senders, function keys, or function key routing slips. You do not need any specific routing parameters because all of your images are placed in the same destination location, with the same field mapping, image processing, and OCR requirements.

Case 2: You want all images from all digital senders to be placed in the same form on a single server, except when you press the ACNTNG function key. When the ACNTNG function key is used, the images must be routed to the accounting storage location (a different folder or database), and a pending form must be created. Your requirement for the ACNTNG function key is the same for all digital senders. In other words, no matter which digital sender is used, if you press ACNTNG, the images must be routed to the accounting storage location.

Case 2 solutions: Follow the same solution as in Case 1, and then add the following: Click the Application icon to add an ACNTNG function key definition to your general routing slip. Change the storage destination and the form name to point to your specific accounting storage location. Because your requirement for the ACNTNG function key is the same for all digital senders, you need to add the ACNTNG function key definition only to the general routing slip.

Case 3: You have the same scenario as described in Case 2, but, you also have three digital senders that belong to the finance department. These digital senders have their own storage location requirements, and all documents that the finance digital senders scan must be routed to the finance storage location regardless of what function key is pressed.

Case 3 solution: Follow the same solution as in Case 2, and then add the following: Create three digital sender-specific entries, one for each of the digital senders in the finance department. The three new entries will have the finance storage location name in their routing slip.

Soft function keys and prompts

Use the digital sender control panel to define soft function keys and prompts. The latest digital sender firmware release supports the creation of prompt fields. The concept is simple: define prompts in the digital sender prompt database, and then attach those prompts to soft function keys. Use the following steps to use digital sender prompts:

1. On the **Digital Senders** tab, select the program that is defined under the digital sender and click the **Define New Key** button to the right.
2. Click the browse button, select a digital sender device, and then select the function that you want to modify.
3. Click the **Modify** button located to the right.
4. Use the **Manage Prompts** tab to gain access to the prompts on this digital sender.

This tab presents a list of prompts that are available on the digital sender that you have connected to, and presents options that you can use to define, modify, or remove any of the prompts from the Digital Sender component.

When all of the prompts are defined, the **Prompt** tab provides a list of available prompts that you can use to add prompts to a soft function key. Select from the available list of prompts, and then click the **Add** button to add these prompts to the function key. After the prompts are added, your digital sender function key prompts you for those values.

Using the Digital Sender 9100c with AutoStore

This example uses the Digital Sender 9100c with Autostore to perform the send-to-folder function. Make sure that you install the digital sender link software on the machine that is to receive the file. This software contains a program (Digital Sender Link), that will create an icon in the service tray in the lower-right portion of your desktop. (In this small interface window, go to the **Settings** tab and select **Show icon** on taskbar.) If you double-click the **Digital Sender Link** icon, a small interface opens. The Digital Sender Link program allows you to name the directory where you want to send the file that you are scanning at the HP Digital Sender 9100c (for example, C:\scanned documents).

NOTE To send a file to a folder, you must install the digital sender link software on the machine that will receive the file.

Start the AutoStore Process Designer (APD). From the **Capture** tab, select and drag the Digital Sender icon to a blank process. From the **Process** tab, select any components that you require. Now go to the **Route** tab and select the Folder Store component.

Double-click the Digital Sender icon to open the **Digital Sender** configuration dialog box. Click the **Digital Senders** tab. The five buttons that appear allow you to configure your process. The following list describes the buttons as they appear, beginning from the uppermost button.

- **Define new Digital Sender.** Click this button to specify a new digital sender. You can type the IP address, or click the browse button to select from a list of available digital senders. A password dialog box might appear. Typically, the username is *Administrator* and the password is blank.
- **Define new Application.** Click this button to name the top-level key that will be used on the digital sender. If you do not have any configured names, click the browse button to select from a list of available configured names.

- **Define new Key.** Click this button to create a new key. This button works similarly to the **Define new Application** button until you get to the **New Function Key** dialog box. Fill in all of the fields in this dialog box. The **Destination Network Address** is the IP address or the HostName of the machine to which you are sending the file. *Do not* fill in the **Application Tag**. Click **OK**.

  Click the **Manage Prompts** tab. The attributes in this dialog box allow you to configure what the buttons will look like under the menu. They also allow you to require users to fill in blanks at the control panel of the Digital Sender. This allows users to customize each scanned document. You can also enter a default value that is assigned when users do not enter any data. Click **OK** and then click the **Prompts** tab. You must highlight all of the keys that you have created and add them.

- In the dialog box, select an item that you want to remove, and then click **Remove Item** to delete that item.

- **Configure Item.** Click this button to see a list of components in your AutoStore process. Select an item, and then click the **Configure** button to configure that component.

Configuring other components

On the **Digital Senders** tab, click **General**, and then click the **Configure Item** button (at the lower-right of the screen) to open the **Configure Components** dialog box. Select the component that you want to configure and then click **Configure**. While you are in the **Configure Components** dialog box, you can type a character in the **Separator** text box to use as a separator between the user name and secret keys. Another use for separators is described in the following paragraph.

Separator

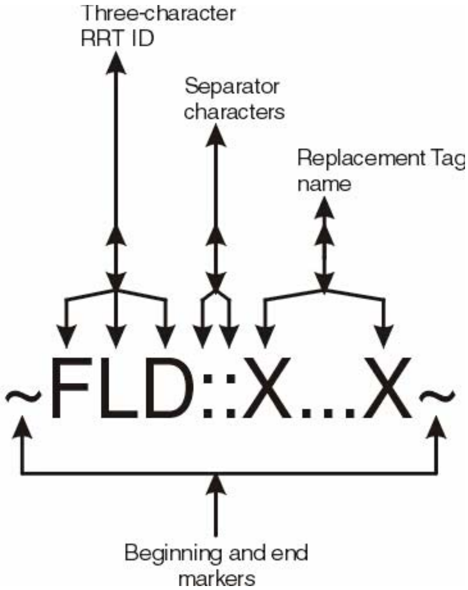
Application Tag is a method of associating a single field as metadata with a document. The administrator can use the Digital Sender component to define a separator character to be used for parsing through this single field and extracting multiple values. For example, if the administrator defines a separator character as " , ", when the user types the value "Invoice , 2004", AutoStore parses through this field and extracts two fields as T1=Invoice and T2=2004.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	**Description**
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of the components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **HDS**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
Version	Version number of the .HPS file.
ScannerName	The name of the scanner as it is published within the .HPS file. This can be the IP address or the host name for the scanner.
ScannerModelName	The name "9100c" as it appears in the digital sender .HPS file.
Sender	The sender's name. Generally, this is the name of the person who has logged into the digital sender. If authentication is not turned on and set up properly, then the default sender name is used.
Title	The file title as it appears within the .HPS file.
ScnSettingType	The values of Scanner Setting Type, represented by a number that appears within the .HPS file.
ScnSettingName	The values of the Scanner Setting Name as it appears within the .HPS file.
Pages	The number of pages, which the device provides within the .HPS file.
Compression	The compression that the device uses, as it appears within the .HPS file.
Format	The file format, coded as it appears within the .HPS file.
Duplex	The scanner duplex or simplex setting attribute within the .HPS file.
Status	The status, at it appears within the .HPS file format.
ApplicationName	The Application (program) Name for the function key that the user presses.
ApplicationItem	The Application (program) Item name for the function key, as it appears within the .HPS file.
ApplicationPath	The full path of the Application (program) and where the files have been stored, as it appears within the .HPS file.
SenderAddress	The Sender e-mail address, as it appears within the .HPS file. A valid user e-mail address requires Authentication to be turned on and active.
NumberOfMenuTags	The number of menu tags that appear for this Application Item within the .HPS file.

Name	Description
MenuTagPromptN	Menu Tag Prompt Name Number "N". This is the prompt number "N" that appears on the front control panel of the digital sender.
MenuTagAnswerN	Your response to menu tag "N". This is the value that you typed in for Prompt "N".
MenuTagDisplayN	The Menu Tag Display Name for "Nth" field on the digital sender screen.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

Troubleshooting tips

Problem	Solution
The .HPS files are processed but the image files remain in the Inbox directory.	This happens when the .HPS file does not point to the image file correctly. This can happen when the Digital Sender Service Inbox directory is not the same as the AutoStore Inbox directory. Digital sender-generated .HPS files contain the path name to the image file within them. If these files are moved but the .HPS file is not updated, then the image file-path name in the .HPS file does not match the new image path name. Either configure the AutoStore inbox to be the same as the Digital Sender Service Inbox directory or correct the .HPS file path name before placing these files in the AutoStore Inbox directory. You can also use a file type of .PDF or .TIF if your image file-type is fixed.
The .HPS files show up on the destination store.	Select the HPS Passthrough option.
An error occurs when selecting the start option on the Service manager.	At a minimum, the Input and Working folders should be available when starting the service. This also applies to the network shares that are used as Input and Working folders. That is, the Input folder is \\MYSERVER\MYSHARE\INPUT\ and the Working folder is \\MYSERVER\MYSHARE\WORK\. Start the service by using a Domain account that has full access to these shares. If the Domain does not have full access, the error message "Cannot Start Service" appears. Put all of the folder path attributes on a local hard drive in order to enhance performance.

Restrictions and limitations

The following restrictions apply to the Digital Sender component:

- Only one Digital Sender component per process can be used.
- The Digital Sender component can only process .HPS/image file pairs.
- The Digital Sender component reads .HPS/image file pairs into the process one pair at a time.
- The Digital Sender Link software must be running on the same machine that the server is running to allow the Digital Sender component to communicate with the server.

ABM Exporter component

Use the ABM Exporter component as a scalable solution for managing your digital sender address books. Use this component to export all of the entries within a device address book to create a batch file that is ready to be sent to other digital sender devices.

The ABM Exporter Capture component and the ABM Importer Route component provide the ability to create primary and secondary relationships and to synchronize the address book entries.

Configure the ABM Exporter component to extract address book entries from a single primary device, and create a secondary device group by using the ABM Importer component. When you run this process, all of your address book entries will be exported from the primary device and imported into the secondary devices. This is the best way to synchronize your digital sender address book entries across an organization. When the two components are set up, an administrator needs to maintain the address book entries and updates on the primary device so that they are automatically propagated to all of the secondary devices.

Feature highlights

Use the features provided by the ABM Exporter component to perform the following tasks:

- Export all or part of your digital sender address book.
- Create a batch file for backup purposes.
- Send exported address-book entries to one or more groups of digital sender devices.

Using the ABM Exporter component

The following case scenarios show typical uses of the ABM Exporter component.

Case 1: Create a backup file of the address book from a primary Digital Sender 9100c in a backup directory, and then use process chaining to route the backup files into other Digital Sender 9100c devices. The processes within the chain are listed here:

Process 1: Export the address book from the Digital Sender 9100c. Export the address-book file entries out of a Digital Sender 9100c device and store them as a batch file into a directory. (ABM Exporter to Folder Store)

Process 2: Load the address-book entries into other devices. Create one or many processes to poll the directory where the batch file is stored and distribute the batch file to many devices. Use the File Option component to capture failed files back to the same Failure directory for reprocessing. (Poll Directory to File Options to ABM Importer)

Case 2: Keep a copy of the batch file and retry importing the address-book entries in case of failure. The processes within the chain are listed here:

Process 1: Export address-book entries from one device, and write into many devices. Use the File Options component to safe keep the batch file in a Reject directory in case of a failure. (ABM Exporter to File Options to ABM Importer)

Process 2: Use the Poll Directory component to check and read the Reject directory from Process 1 and resend to the Digital Sender 9100c on a periodic basis. Use the File Options component to capture failed files to the same Failure directory for reprocessing. (Poll Directory to File Options to ABM Importer)

Configuring the ABM Exporter component

The following attributes are available in the **ABM Exporter** configuration dialog box:

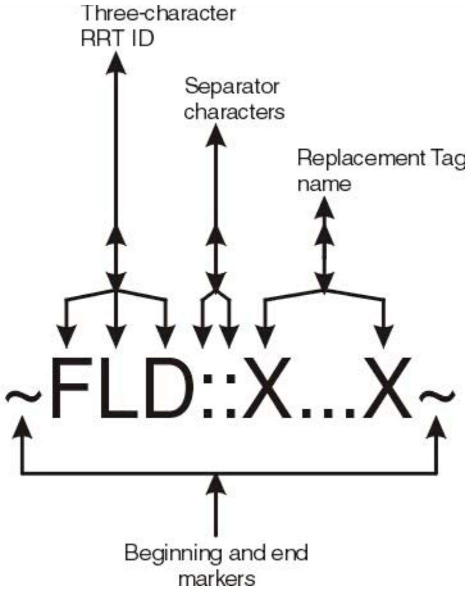
- **Name or IP Address.** Type the host name or the device IP address of the primary device from which you want to export address book entries.
- **Password.** Type in the administrator password for the selected primary digital sender. If the password is blank, leave this field blank.
- **Export Address Books.** Select from a subset of address-book entries.
- **Public Email.** Select this check box to export public e-mail address book entries, excluding the distribution lists.
- **Public Email Distribution Lists.** Select this check box to export the the e-mail distribution lists.
- **Public Fax.** Select this check box to export the public fax entries out of the fax database.
- **Public fax distribution lists.** Select this check box to export the fax distribution list entries.
- **User Profiles.** Select this check box to export user profiles to the batch file.
- **Private Email.** Select this check box to export private e-mail entries from the user's private address book.
- **Private Distribution Lists.** Select this check box to export private distribution lists from the user's private address book.
- **Printers.** Select this check box to export the printer definitions.
- **Function Keys.** Select this check box to export the function key definition. This feature is designed to support the function keys that older versions of the digital sender firmware support.
- **Function Keys (new style).** Select this check box to export all of the function keys that the newest version of the digital sender firmware supports.
- **Prompts Choices (new style).** Select this check box to export the prompt field definitions. Only the newest version of the digital sender firmware supports this feature.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The ABM Exporter component does not generate RRTs.

Troubleshooting tips

Problem	Solution
The ABM Exporter component does not produce any batch files.	Make sure that the administrator password matches the administrator password on the digital sender device. Make sure that the IP address of the digital sender device has not changed.

Restrictions and limitations

- Use the ABM Exporter component with a digital sender.
- The ABM Exporter component generates a batch file that can be stored as a text file or used as input into ABM Importer.
- The ABM Exporter component has no limit on the number entries in any of the address-book entries.

Knowledge Package Loader component (Capture)

Use the Knowledge Package Loader component to read .XML files. You can use the XML schema to encapsulate index fields and files, and to communicate batch job attributes to various software programs.

This component uses the AutoStore XML schema.

Feature Highlights

Use the Knowledge Package Loader component to perform the following tasks.

- Load .XML files into an active AutoStore process.
- Encapsulate document or image files.
- Encapsulate unlimited index fields.
- Use static or dynamic field values.

The Knowledge Package Loader component is a blocking component. To add components to the process, click the **Components** tab, and then select the component that you want to add. Click **Configure** to configure that component.

Using the Knowledge Package Loader component

Typical uses for the Knowledge Package Loader component include the following tasks.

- Polling a directory by specifying the Input and Working directories. Make sure that the Input and Working directories are different so that you do not create a never-ending process.
- Removing or storing files based on the success or failure of a process.
- Determining whether or not to include the field values of the .XML file that is being processed.
- Configuring other components within a process.

The Knowledge Package Loader component is typically used with the PDF417 Barcode component. The most common input types are .XML files specified by AutoStore schema.

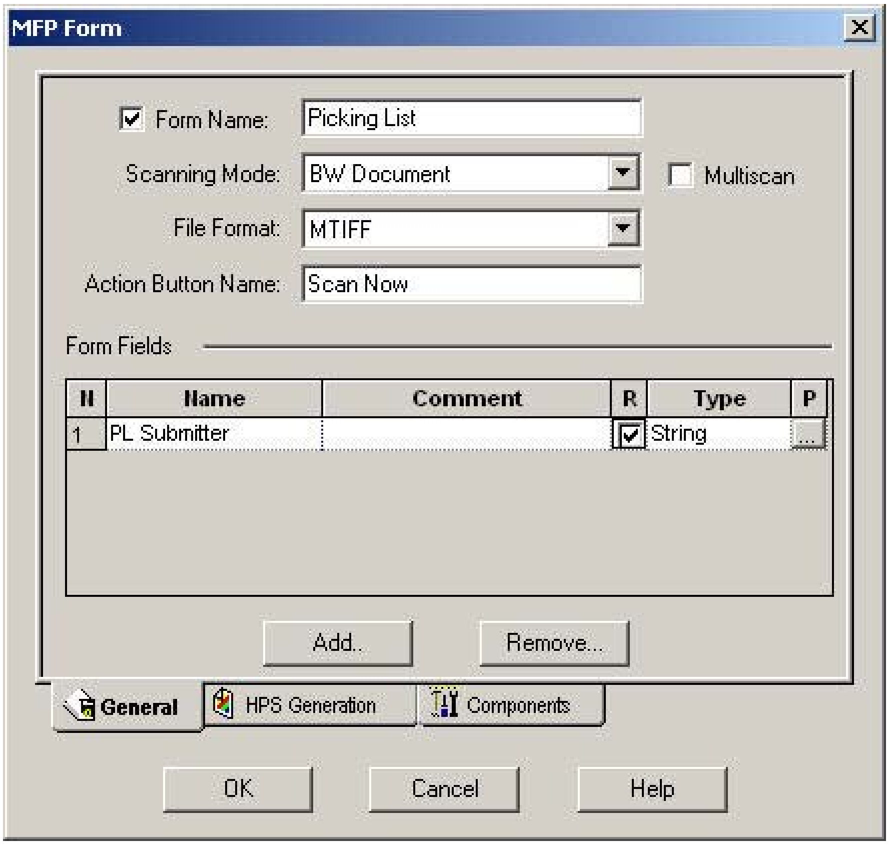
NOTE Do not place a Knowledge Package Loader Process component immediately after a Knowledge Package Loader Capture component unless you ensure that the Capture component is going to generate an .XML file to be processed by the Process component.

The Knowledge Package Loader component is a blocking component. To add components, click the **Components** tab. Select the component that you want to configure, and then click **Configure**.

NOTE Two Capture components cannot be used within one process. The Working directory and the Input directory that are configured in the Knowledge Package Loader component should be different.

The following examples describe how to use the Knowledge Package Builder and Knowledge Package Loader components with an MFP to encapsulate information into the .XML data format and to extract information from the .XML data format. The benefit of using the .XML data format in AutoStore processes is that the format can contain image data and also store user-defined information that can be retrieved when the data is scanned.

1. Load an AutoStore configuration file into an MFP that contains form fields and schema data that can be modified by using the MFP.



MFP Form

☑ Form Name: Picking List

Scanning Mode: BW Document ☐ Multiscan

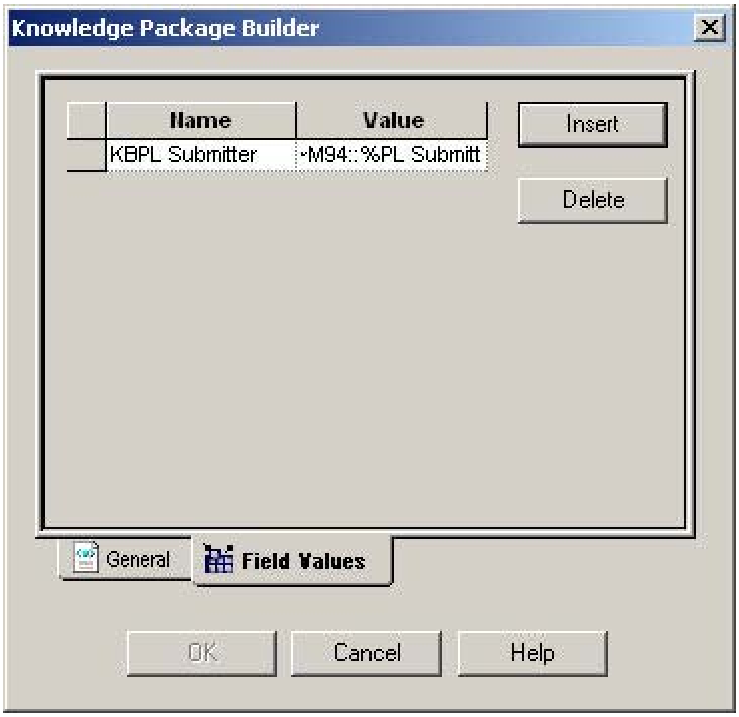
File Format: MTIFF

Action Button Name: Scan Now

Form Fields

N	Name	Comment	R	Type	P
1	PL Submitter		☑	String	...

The configuration file also contains a Knowledge Package Builder component that is used to generate the .XML data. The administrator has added form fields within the Knowledge Package Builder component that capture the data from the MFP forms.



Name	Value
KBPL Submitter	~M94::%PL Submitt

2. At the MFP, use the AutoStore **Send To** menu to enter data for the form fields that were generated by the AutoStore configuration file. When you are finished, press **Scan Now**, which is the **Action Button Name** that was designated in the configuration file.

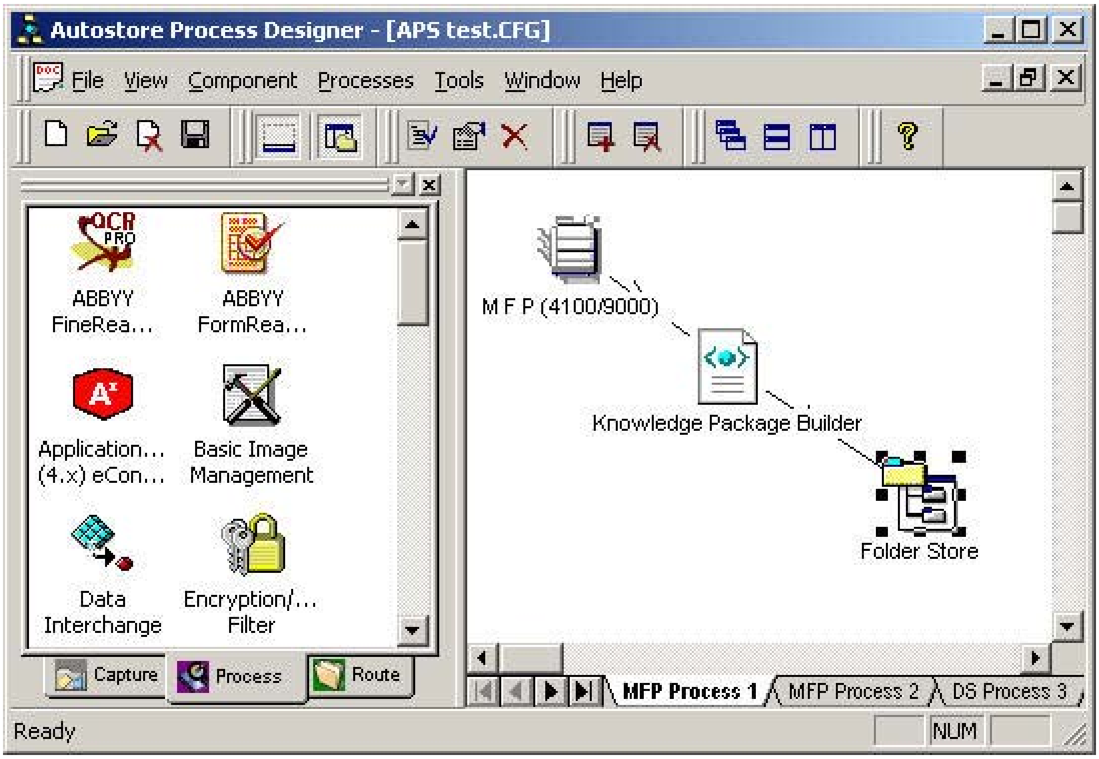
3. The MFP collects the MFP user-specified information as .XML metadata, scans the image, and then sends all of the information to the AutoStore server.

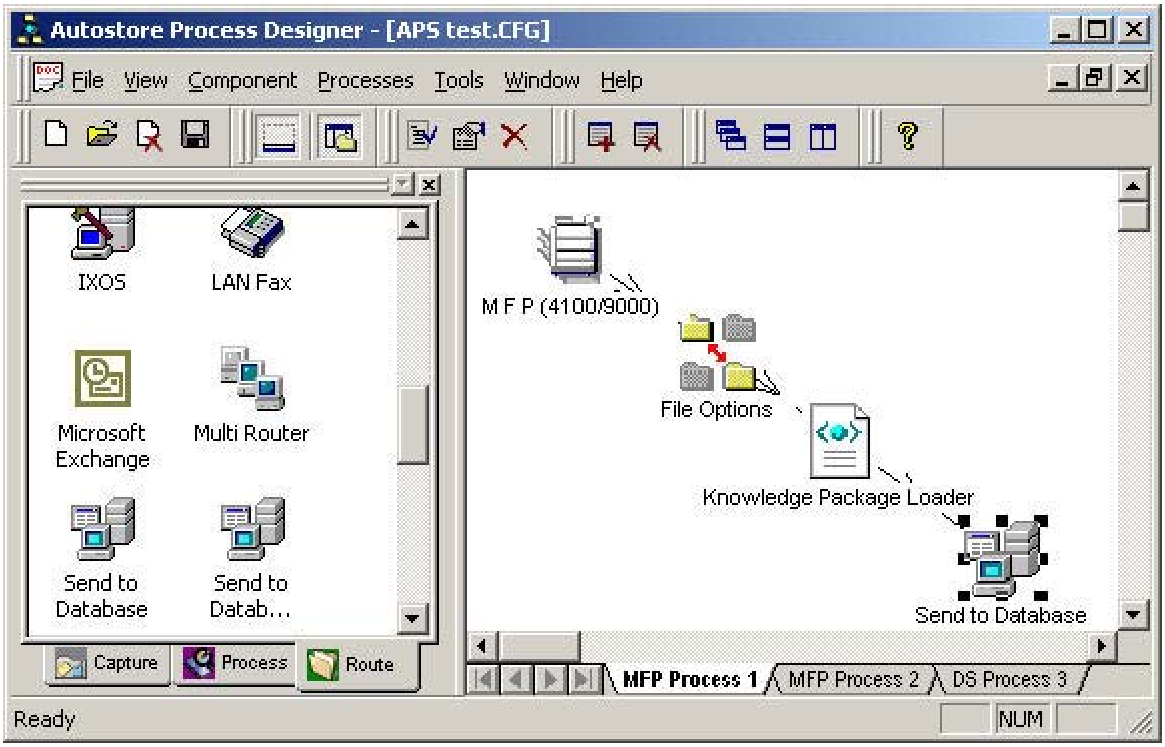
4. The AutoStore server starts the Knowledge Package Builder component, combines the metadata information and the image information into a single .XML file, and then pushes this information through the rest of the AutoStore process.

In a chain process, or on a different AutoStore server, the AutoStore administrator sets up additional process configuration tabs that contain a Poll Directory component that collects the .XML file and feeds the file into the Knowledge Package Loader component.

The Knowledge Package Loader component separates the metadata from the image. At this point, the metadata that was captured at the MFP as user-specified information is made available, along with the image, to the new AutoStore process.

The following two images are examples of the process and chain process that use the Knowledge Package Builder and Knowledge Package Loader components.





The metadata information is collected from the MFP in the form of the Submitter Name (PL Submitter). Through the AutoStore process, the metadata information is processed in .XML format and sent to the Chain process so that the Knowledge Package Loader component can extract and decode the .XML data and make the metadata available to the rest of the chain process. The metadata can be stored in a database.

The Knowledge Package Builder and Knowledge Package Loader components allow you to collect user-defined metadata from an MFP at the time of an image scan. Then, you can use this metadata in other chain processes, within the AutoStore server, or other knowledge object software applications. Without these two components, the metadata is lost after the first process completes.

Configuring the Knowledge Package Loader component

Use static or dynamic values that are defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Knowledge Package Loader component.

The following attributes are available in the Knowledge Package Loader dialog box.

General tab

Use this tab to set the following attributes:

- **Input Directory.** Select the directory from which you will poll for files. Click "**...**" to browse a list of available directories.
- **Working Directory.** Select the directory where you will place files temporarily while they are waiting to be processed.
- **On Success.** Select one of the following actions for the component to perform when the execution is successful:
  - **Remove Files.** Select this check box if you want to remove the files from the process when execution is successful.
  - **Move Files.** Select this check box and specify the path and folder where you want the files to be moved when the execution is successful.
- **On Failure.** Select one of the following actions for the component to perform when the execution is not successful:
  - **Remove Files.** Select this check box if you want to remove the files from the process when execution is not successful.
  - **Move Files.** Select this check box and specify the path and folder where you want the files to be moved when execution is not successful.
- **Include Fields.** Select this check box to include the field values from the knowledge object into the AutoStore-specified XML schema. When this check box is selected, the .XML field values are transferred into the next component if that component can accept field values. For example, consider a scenario where the next component is a Send to Database component with an Access table set up with the same Field Names as those in the processed .XML file. If the **Include Fields** check box is selected, then the Knowledge Package Loader Process component automatically sends the .XML field values into the Send to Database component.

Components tab

Use this tab to set the following attributes:

- **Component Name.** View the names of the components that are currently available in your process. You can click to select a component, and then click the **Configure** button to configure that component.
- **Configure.** Click the **Configure** button to configure the selected component.

Using the Knowledge Package Loader Capture component in a multiprocess chain

A multiprocess chain is a set of processes in which output from one process feeds into the input of another process. Process chains are useful when you have images or data elements that must be routed to multiple destinations. For example, a process that requires routing to SharePoint Portal Server, Microsoft Exchange, and Send to Mail Recipient must be designed as a three-process chain. The first process routes information to the SharePoint Portal Server and, on success, the process stores the files in a folder destination for use in a second process. The second process uses the Poll Directory Capture component to grab the files from the directory and route them to Microsoft Exchange. The third and final process uses the Send to Mail Recipient Route component to route the same files by e-mail.

The following example describes how to use the Knowledge Package Loader Capture component in a multiprocess chain.

After using the Knowledge Package Loader to capture .XML files, you want to store the files in a Success or Failure directory. You then want to use the Success directory to send the files to additional destinations, and use the Failure directory to further process the failed files.

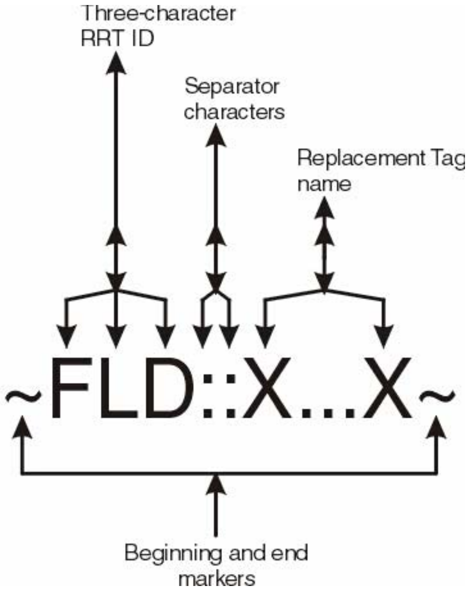
In the first process in the chain, AutoStore reads .XML files from a directory and sends them to Destination A by using the specified Route component. The Route component sends the files into either a Success directory or a Failure directory. In the next process, AutoStore uses the Poll Directory Capture component to send the files in the Success directory to Destination B by using the specified Route component for this secondary process. For the files in the Failure directory, AutoStore uses the Poll Directory component to send the files to a different folder, as specified by the Route component (such as Send to Mail Recipient) in this secondary process.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is ASX.

The following table describes the Reserved Replacement Tag Names (RRTN) values for this component.

Reserved Replacement Tag Names (RRTN)

Name	Description
Version	The version number of the XML schema.
DateCreated	The date that the .XML file was created.
TimeCreated	The time that the .XML file was created.
AUTHOR	The field where you type the author's name.
COMMENTS	The field where you type comments into the file.

Field Replacement Tag Names

This component does support Field Replacement Tag Names (FRTNs) for field names contained within the .XML file. The following is an example on an FRTN.

~ASX::%Client ID%~ is replaced by "Hewlett-Packard" if the user types in `Hewlett-Packard` for the field name "Client ID."

Special Set Replacement Tag Names (SSRTN)

No SSRTN is supported by this component.

Troubleshooting tips

Problem Description	Solution
You cannot configure a component because it does not appear on the list of components on the **Component** tab.	You must add the component to the AutoStore process or it does not show up in the Knowledge Package Loader blocking component.
An error message appears when you start the AutoStore Service Manager.	Make sure that the Input directory exists.
A "Duplicate Output destination" error appears when you attempt to run your AutoStore process.	If the **Include Fields** check box is selected and the next component in the process (for example, Send to Database) is using RRT ASX, the process will fail with a database error.

Restrictions and limitations

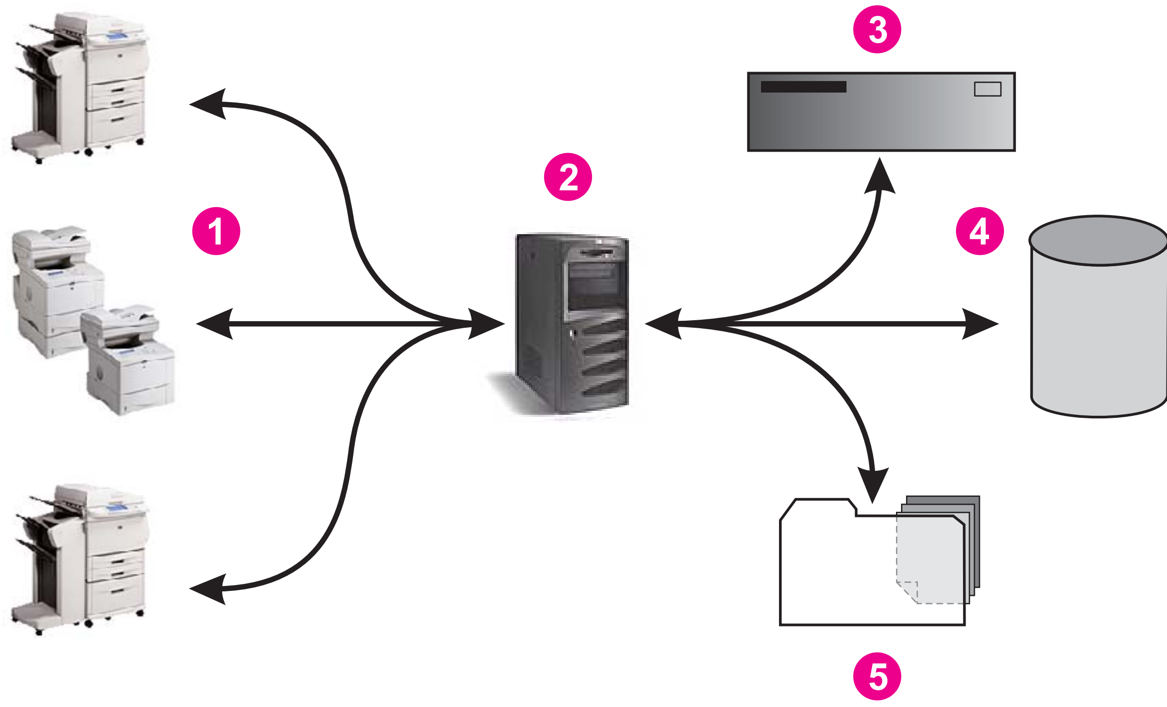
- Do not make the Input directory the same as the Working directory.
- Make sure that the Input and Working directories are valid.
- Make sure that the Success and Failure directories that you select are valid.
- Do not place a Knowledge Package Loader Process component immediately after a Knowledge Package Loader Capture component unless you ensure that the Capture component is going to generate an .XML file to be processed by the Process component.

MFP (4100/9000) component

The MFP component is designed to work with Chai .JAR-enabled HP devices and can communicate with these devices in following ways.

- Create and update application function keys by using XML
- Receive scanned document and indexing data
- Provide service-based send-to-e-mail functionality

The following illustration shows the AutoStore server integrated as the middleware software for MFP devices.



1. **MFP devices**
2. **AutoStore server**
3. **E-mail and fax devices**
4. **Software programs**
5. **Folder Store, FTP Store, Send to Printer, and Send to PC components**

Use the MFP (4100/9000) component to administer all of the function-key capabilities of the HP Chai .JAR-enabled devices.

Different MFP models have different control-panel layouts, and contain either a keypad or a touch screen.

Both the keypad and touch-screen control panels have some functions that are similar.

- The **Start**, **Stop**, and **Reset** buttons control operation.
- The Ready, Data, and Attention lights show status.
- The numeric keypad is used to type numbers.

The MFP (4100/9000) component communicates with Chai .JAR files that are installed on the MFP devices. The device-to-server communication attributes are controlled directly from the Chai Servlet embedded Web server interface that is installed on the device.

The MFP (4100/9000) component receives electronic images and index data from all devices and initiates the correct processes to transfer the electronic data to the designated destinations. The configuration parameters in the MFP (4100/9000) component let the server make decisions about how to route the electronic images as e-mail or capture the electronic file according to the function-key definitions.

The MFP front panel interface can be used for the following tasks:

- **Application function keys.** Press the **Menus** button on the front panel and then use the function keys that appear to guide you to the correct form.
- **Index Documents.** Use the indexing fields to provide index attributed for documents.
- **Scan Documents.** Press an action button to send documents and index data to the server.

You can create and manage all of your device menu entries directly from this screen. Use the buttons on this page to add, edit, and modify the menu entries and corresponding routing slips for each form.

Feature highlights

The MFP (4100/9000) component offers the following basic features:

- Create and manage function keys on the MFP front panel.
- Create indexing forms and fields for MFP devices.
- Create and administer logical device groups.
- Manage the SMTP gateway and e-mail functionality directly from the device.
- The MFP (4100/9000) component includes dynamic routing of electronic documents within a single process by using the Multi Router component.
- Use the Send-to-PC capabilities for MFP devices to send files directly to computers.

The MFP (4100/9000) Add-on (Enterprise) license options offer the following enterprise add-on features:

- This component provides Server Farm support, which allows multiple AutoStore servers to service all of the MFP devices within a network. Use this feature for geographical site load balancing routers between the MFP devices and the servers.
- HP 9100c Authentication server for authentication across hybrid environments for Windows Active Directory and LDAP servers.

NOTE The AutoStore License Manager controls the enterprise add-on features.

The MFP (4100/9000) is fully integrated with the following items:

- Local MFP Fax address book that the administrator can use to create a user field that is linked into the MFP local fax speed-dial address book. Use this feature to create LAN Fax-enabled speed-dial buttons that are directly linked into the local address book of the MFP device. This provides a way in which the AutoStore MFP (4100/9000) component can access the local Fax address book on an MFP. A user can then view and select preconfigured fax-enabled speed dial buttons from an AutoStore menu. This can be used when addressing LAN Fax jobs when the MFP (4100/9000) component is used with the LAN Fax Route component.
- Local e-mail address book that can be used to create function keys that can include the option to select e-mail addresses from the preconfigured local address book on the MFP.

The MFP (4100/9000) component requires MFP Enterprise Licensing. This special licensing permits the MFP enterprise enhancement.

Setting up the MFP (4100/9000) component

Use the following steps to complete your MFP setup:

1. Follow the MFP installation procedure to install the required .JAR files on your MFP device.
2. Make sure to follow the installation procedure to configure the AutoStore Chailet on the embedded Web server of the device.
3. Use the AutoStore Process Designer to create an AutoStore process and use MFP as the Capture component.
4. Configure the MFP component for the following actions:
   - Use the **MFP Menu** tab to create menus, forms, and indexing fields for the Common Group or any device- specific group(s).
   - Use the **SMTP Gateway** tab to set up the external SMTP server for routing the e-mail messages.
   - Use the **Preferences** tab in the MFP Component to set up the Working directory parameters.
5. Use the **Components** tab on each form to configure all of the Process and Route components.
6. Save all of the configuration files in a .CFG file.
7. Start the AutoStore Server by opening the .CFG file in the AutoStore Service Manager and selecting **Apply**, and then clicking **Start**.

MFP component FAQs

Question	Answer
Can I create two AutoStore Processes that use the MFP component?	No, you can only create a single AutoStore process that use the MFP component as the Capture component.

MFP component FAQs (continued)

Question	Answer
Does the IP Port on the **Preferences** tab have to match the IP Port that is configured on the AutoStore configuration page on the device?	Yes, the MFP Device Chai Servlet and AutoStore use the same IP port (that is, 3232) to communicate, and the ports must match the AutoStore IP Port number on all of the devices.
Do I have to configure Menus and Forms on the MFP component to get menus and forms to appear on the front panel of the device?	Yes, all of the function keys are centrally controlled from the AutoStore MFP component.
I configured all of the menus, forms, and index fields on MFP component but they do not appear on the front-panel menu list on the MFP. What could be the problem?	1. Make sure that you save your .CFG file and start your AutoStore server by using the correct .CFG file. 2. Make sure that the device is configured to use this AutoStore server by checking the AutoStore server IP address in the AutoStore URL in the embedded Web server of the device. 3. Make sure that the device AutoStore IP Port matches the AutoStore IP Port (on the **MFP Preferences** tab). 4. Use the AutoStore URL **Update Now** button so that the device will get the latest menu buttons.

Using the MFP (4100/9000) component

You can use the MFP component to perform the following tasks:

- Capture files from MFP devices and route the files to a variety of destinations, such as Folder Store, FTP Store, DMS, or other components.
- Capture index data and populate index fields into programs.
- Create function keys on supported HP MFP devices.
- Create groups of HP MFP devices and maintain function keys that are based on the device group.
- Capture and route SMTP e-mail messages.

Configuring the MFP (4100/9000) component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the MFP (4100/9000) component.

The MFP configuration consists of three separate tabs:

- **MFP Menus**. Construct menus and configure routing attributes.
- **SMTP Gateway**. Create a prioritized list of SMTP Gateway servers.
- **Preferences**. Establish settings for process home directory, IP Port Number, and other administrative directories.

MFP Menu tab

The MFP **Menu** tab consists of a single graphical display that contains all of the MFP device groups, menus, forms, and fields. Each of these entities can be managed directly from this tab by using the buttons that are available on this tab.

The MFP **Menu** tab contains at least one entry for the most common function key definitions. This entry is represented by the “Common MFP Group” that appears at the top of the list. The Common MFP Group, unlike other MFP groups, does not take any device members. By default, any MFP that is not a member of other MFP groups is considered to be a member of this group.

Use the Common MFP Group to create the default menus and forms definitions that appear on your most frequently used MFP devices. If this group-definition schema is used, new MFP devices that are added by installing Chai .JAR files do not have to be configured on the server. Any new MFP is automatically assigned the common function-key definitions.

- **Add Group.** Use this feature to add a device group. The device group must have a minimum of one device member. Each group entry requires the following entries:
  - **Group Name.** A string value describing the name of the group. This name does not appear anywhere on the MFP device, and it is used strictly as a logical group name.
  - **Root Menu Title.** The action button that will appear as the first choice on the MFP control panel. Use an action term such as "Send Files" or "Scan to" to name this button. You should be able to immediately relate this front-panel button with the transmission of electronic documents.
  - **Group Devices.** Use the following buttons to manage the entries on this list:

    Add. Add a device to the list. You can use the device network name or the device IP address. It is recommended that you use a network name to accommodate possible changes in IP addresses when using the dynamic host configuration protocol (DHCP).

    Remove. Click on a device entry and press **Remove** to remove a device from the list.
- **Add Menu.** Add a menu to create a function-key hierarchy. The menu items are entries within a function-key-tree hierarchy.
- **Add Form.** Add a form to capture index data and describe the processing attributes. To add a form, click the **Add Form** button and fill out the **General**, **HPS Generation**, and **Components** tabs.

The following table describes the information that is required on the **General** tab.

General tab

Field name	Description
Form Name	The name of the form that appears as a button name on the MFP function-key hierarchy.
Scanning Mode	Select to scan a black-and-white document or color document.

General tab (continued)

Field name	Description
Multiscan	Selecting this value will prompt the user after each scan job for additional pages to be scanned. Note that this option does not provide the capability to switch input feeder between the flatbed and automatic document feeder for a job. With flatbed multiscan, you can scan multiple documents of the same or different sizes. With automatic document feeder multiscan, you can scan different sized, and large numbers of documents.
File Format	Select one of the following options from the drop-down list: ● JPEG ● MTIFF ● PDF
Action Button Name	The button name that appears as the last key on the form. Use a descriptive action name such as "Scan" or "Scan to" to indicate its function.

General tab (continued)

Field name	Description
Form Fields	Add fields by clicking the **New field** button that appears as the first column header and is labeled with "N". When you have added the first row, you can use the **Tab** key to move through and add additional fields. For each field, configure the field Name, Comment, and Type. The following are the field attributes: ● **Field Name:** The field label name is a string type and consists of any alphanumeric characters. Use of special characters (~, !, @, #, $, %) is not recommended. **NOTE** Do not use **Sender** or **SenderAddress** as field names. These are predefined as special case FRTNs. ● **Comment:** The description field for the administrator's reference. This field is not currently used anywhere within the MFP. ● **Type:** – String: Alphanumeric field – Integer: Integer values – Float: Floating numbers – String List: Keyword list of values – Email Lookup: E-mail search contains addresses from the MFP local address book – Fax Lookup: Fax-number search contains numbers from the MFP local address book ● **Required:** Select this check box if you want the field entry to be mandatory. For each field type, you can set additional field attributes by clicking the **Properties** button in the right column of each entry.

The following table describes the information that is required on the **HPS Generation Tab** tab.

Field name	Description
Generate HPS	This enables .HPS file generation.
Title	This string becomes the Title field within the .HPS file.

Field name	Description
Destination	This is the folder path where the .HPS folder and image file are going to be stored.
Passthrough	When this field is enabled, the image file can be processed as it is configured within the **Component** tab. When this field is not enabled, processing stops after generating the .HPS and image files. The files are not processed by the remaining components within the process. Turn Passthrough “off” when all you need is an .HPS or image file and you do not require files to be processed by the remaining components.

NOTE The .HPS file does not support .JPEG file format. When you create a button that generates a .JPEG file, no .HPS file can be generated.

The **Components** tab lists all of the Process components, their description, type, and the configure button to set their configuration attributes. The configure button appears as the last column within each row with **...** appearing on it. Select each component by clicking the component, and then configure the component for this form by clicking on the configure button (last column on each row).

SMTP Gateway tab

The SMTP gateway is a required part of the MFP component configuration. The MFP (4100/9000) component uses the gateway list to route e-mail messages to the configured gateway.

The e-mail servers are listed by priority orders. The MFP (4100/9000) component attempts a connection to each server, beginning at the top of the list.

- **Add.** Add an SMTP gateway by clicking the **Add** button. You can add SMTP gateways by using the server IP address or the server host name.
- **Edit.** Select any SMTP gateway and click the **Edit** button to modify the SMTP server address.
- **Remove.** Select any SMTP gateway and click the **Remove** button to remove the gateway from the prioritized list. The system will ask for confirmation before removing the entry. You **cannot** undo this operation.
- **Test.** Select any SMTP gateway and click the **Test** button to perform a test. A test consists of opening the SMTP gateway port to the server and sending a test message to the server to ensure a valid SMTP gateway is present. A message that confirms the status of the gateway appears.
- **Move.** Use the **Move** button to modify the priority list order. The AutoStore server attempts to use the prioritized list of SMTP gateways in the order that they appear on this list. The top SMTP gateway is considered to be the first used and the final entry is the least likely to be used. Click on any SMTP server and change the order in which it appears by using the up and down arrows.

Preferences tab

The **Preferences** tab provides the basic attributes that control how MFP messages are stored, forwarded, and routed.

- **Home Directory.** This is the root directory where AutoStore creates all temporary directories and files that are necessary for controlling the MFP-related traffic. The AutoStore server must have adequate permission to write to this directory.
- **Port Number.** This is the IP port number that is used between the MFP device and the AutoStore server for communicating the Chai servlet configuration attributes. This port number **must** match the port on the Chai servlet.
- **Keep.** To determine where your files are stored, select one of the following check boxes:
  - **Processed Files.** The Processed Files directory is where the captured files (send to folder or program files) are stored upon the completion of successful routing. If you turn this attribute on, all files are stored to the Processed Files directory when AutoStore successfully completes writing them to the destination location.

CAUTION If this is turned on, a copy of every file that has been successfully routed is saved to the Processed Files directory. Make sure that adequate disk space is allocated to this directory.

  - **Rejected Files.** The Rejected Files directory is used to store any files that failed to store to the final destination location. The failure could result from any number of reasons, and the failure must be researched by reviewing the program log file entries. Typically, HP recommends having a secondary process that takes this directory as an input and routes messages to the system administrator to warn about the failure. AutoStore must have sufficient rights to access this directory.
  - **Rejected eMails.** The Rejected eMails directory is used to store any outgoing e-mails that the receiving SMTP server has rejected as undeliverable. Any SMTP server rejection forces AutoStore to reject the e-mail message. AutoStore does not attempt to use any subsequent SMTP server after any one SMTP server explicitly rejects the message.

Using the MFP (4100/9000) component in a multiprocess chain

A multiprocess chain is a set of processes in which output from one process feeds into the input of another process. Process chains are useful when you have images or data elements that must be routed to multiple destinations. For example, a process that requires routing to SharePoint Portal Server, Microsoft Exchange, and SMTP e-mail must be designed as a three-process chain. The first process routes information to the SharePoint Portal Server and, on success, the process stores the files in a folder destination for use in a second process. The second process uses the Poll Directory Capture component to grab the files from the directory and route them to Microsoft Exchange. The third and final process uses the SMTP Route component to route the same files by e-mail.

The following case illustrates how to use the MFP (4100/9000) Capture component in a multiprocess chain.

NOTE Currently, the Processed files or Rejected files attributes of the MFP (4100/9000) component cannot be used to directly create a multiprocess chain because it writes both failed and successful files in .EML format. Currently, no other AutoStore component can read files that are in .EML format. However, you can use the Knowledge Package Builder component to convert files to .XML, and then use the Knowledge Package Loader component to chain the processes. Case 1 shows an example of this type of multiprocess chain.

A multiprocess chain is used to implement the storage of files from one Capture component into multiple Route components. As the MFP (4100/9000) component generates an .EML file, in order for the .EML file to be chained to other processes, you must have a Capture component within the next process chain that can read and understand the content of an .EML file.

At this time, no components can receive an .EML file as input, so to create a process chain for the MFP (4100/9000), you must first convert your .EML file into an .XML file (using Knowledge Package Builder), store the .XML file into a folder (using Folder Store), and use the Knowledge Package Loader Capture component to create your process chain or chains. Design your processes as described in the following procedure:

First process. Use MFP (4100/9000) capture process, followed by Knowledge Package Builder and the Folder Store Route component. The Knowledge Package Builder in this process creates an .XML file from the Knowledge Object (file and metadata) that can be used in the subsequent process chains.

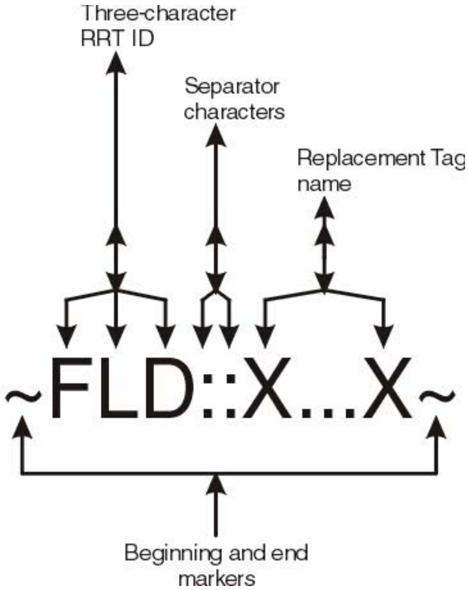
Subsequent processes. Design process chains with a Knowledge Package Loader Capture component to read the .XML file from the Output directory of the first process and route the content to the appropriate Route component. Set the Success folder of your Knowledge Package Loader to point to the input of subsequent processes within the chain. You can create as many processes as you require for routing the .XML file to all of the Route components that you want.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure is an illustration of the RRT definition. Each part of the RRT creates a standard field tag replacement that can be expanded to all of the components that are developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT:

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **M94**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
PageCount	The number of pages that have been received.
Format	The format values: 5 is PDF, 2 is MTIFF, 3 is JPEG.
FileSize	The received file size.
IP	The IP address of the MFP device.
MainMenu	The top level menu item name.
SubMenu	The sub-menu path.
HostName	If available, this is the hostname of the MFP; if not, this is the IP Address.

The following are examples of RRTNs:

~M94::PageCount~ - replaced with the value "10" for a ten-page document.

~M94::Format~ - replaced with the value "5" if the file that the MFP sends is a .PDF file.

Field Replacement Tag Names

This component supports Field Replacement Tag Names (FRTNs) and replacement of field names that the MFPs generate. The following are examples of FRTNs:

Example 1: ~M94::%Client ID%~ is replaced by "Hewlett-Packard" if someone types `Hewlett-Packard` for the field name "Client ID".

If authentication has been turned on for the MFP, then **~M94::%Sender%~** contains the domain and user name value for the authenticated user. **~M94::%SenderAddress%~** contains the e-mail address for the authenticated user.

Example 2: For form-field types **Email Lookup** and **Fax Lookup**, the field name + "**_name**" contains that entry name in the address book. For example, the local e-mail address book for the MFP contains an entry for **John Doe** with the e-mail address **jdoe2131@hp.com**. You have configured a form field called **email** that is of type **Email Lookup**. When someone uses the MFP to select John Doe from the drop-down list of values, the FRTN **~M94::%email %~** contains the e-mail address for the entry (jdoe2131@hp.com). The FRTN **~M94::% email_name%~** contains the name of the selected entry (John Doe).

Example 3: Define a field on the MFP for sending jobs to a printer using the Send to Printer component. The field definition used to depict the number of copies can be added to your form as a field such as Copies. Within the Send to Printer component, you can use **~M94::% Copies%~** to print the required number of copies of the scanned document on the designated printer.

Special set replacement tag name (SSRTN)

The MFP (4100/9000) component supports the Date/Time field names that appear in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as first day of week (00 to 53)
%w	The weekday as decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

Troubleshooting tips

Problem	Solution
AutoStore rejects some jobs.	The Form or Action button names cannot contain the following characters: @, $, &, <, >, ?, /, \, or |.
The menus on the MFP are populated but no files are being received. Also, the Data LED blinks continuously. No entry appears in the log, because this is not a server issue.	Check the SMTP gateway on the MFP and make sure that the gateway address is pointing to the AutoStore server. The SMTP gateway address on the device appears under the Digital Sending options on the MFP. This address must be set to the AutoStore server.

Problem	Solution
During runtime, the menus are not updated. No "Menu updated" message appears in the log for the device. **NOTE** For each device that is connected to AutoStore, a message that indicates that the menus are being updated should appear periodically.	Check the IP Address in the AutoStore Chai Servlet and make sure that it is pointing to the AutoStore computer. Correct the IP Address in the AutoStore Chai Servlet so that it is pointing to the AutoStore computer. Also make sure that the menu update IP Port ID matches the IP Port on the AutoStore server.
E-mails that the MFP sends do not go to the external e-mail servers.	The Rejected e-mail folder has .EML files. This indicates that AutoStore has received an error when attempting to send SMTP e-mail messages to the mail server. ● The SMTP gateway addresses on the server are not correctly configured to point to the external SMTP e-mail server that relays messages. Correct the SMTP gateway e-mail addresses on the AutoStore server. The Retry Mail File successful e-mail folder has .EML files. This indicates that the AutoStore has tried sending to the SMTP servers but the servers have not been available. ● The SMTP gateway server is not functioning correctly and AutoStore will try again. Check the SMTP server to make sure that it is functioning correctly.
The MFP device is reporting that the e-mail server is not available. The AutoStore logs do not show any errors, because this error is occurring on the MFP.	Check the SMTP gateway address on the MFP device and make sure that the address is pointing to AutoStore server, and that the AutoStore service on that computer is functioning correctly. Check and make sure that the AutoStore server is running.
Only one document appears in the Output directory, and the document is called "DOCUMENT.TIF" or "DOCUMENT.PDF". No error appears in the log. The operation has been completed successfully.	The "Overwrite existing file" option has been set on the Folder Store component. All files that are received from MFP devices are called "DOCUMENT.X", and, unless you choose to rename the files, only one file is created and rewritten. To save files that retain separate names, turn the overwrite option off and create a renaming schema.
AutoStore starts and immediately stops. No error messages appear on the log.	Make sure that IP port 25 is free, both inbound and outbound. Use a command prompt to perform the function "telnet 127.0.0.1 25". If the session opens with any SMTP server name, then the port is already being used. Stop the e-mail service that is using Port 25.

Problem	**Solution**
The AutoStore server does not start. The log entry shows the following message: "Can't create directory."	The folder paths that appear on the **Preferences** tab must all exist. The AutoStore server attempts to create these folders, but if it is not successful or if the folder is deleted before the AutoStore server starts, then the server will not start correctly. Make sure that the folders on the **Preferences** tab exist, and that the AutoStore user has full access to them.
The MFP device does not show the Send folder under the main menu. The main folder actually shows the AutoStore folder.	1. The MFP device is not pointing to an AutoStore server. Check the MFP device configuration and make sure that the Chai Servlet is pointing to the correct AutoStore server. 2. The MFP device is initializing and has not contacted the server yet. The Chai Servlet initializes after initialization is complete. When initialization is complete, the menus should appear on the screen. 3. The AutoStore server does not have a process that contains the MFP (4100/9000) Capture component. The AutoStore server must have one process that includes the MFP device. The MFP process must be configured correctly.

Problem	Solution
E-mail (.EML) files appear in the Rejected e-mail directory.	The external SMTP gateway (the SMTP gateway that appears in the list for the AutoStore MFP) is a valid SMTP server and it is rejecting incoming mail. This could result from the following factors: ● The e-mail address is not a valid Internet e-mail address. ● The SMTP gateway has no way of handling the message, and is therefore rejecting it. Look in the AutoStore log files to find the error code that the SMTP gateway returned, and then contact the SMTP server administrator. To gain access to the rejected e-mail, you must use Outlook express to open and manually correct the issue for each e-mail message. When the messages are corrected, use Outlook express to send the messages again. In order to be notified of any e-mail failures, you must create a process that has the Rejected e-mail directly as its input and SMTP Send as its store. Configure this process to send the administrator an e-mail whenever an .EML file appears in this directory. Because many possible causes exist for this type of failure, no automatic solution to e-mail failures is available.
Files are appearing in the HOME DIRECTORY \OUT QUEUE.	The MFP (4100/9000) component is attempting to contact outgoing SMTP gateways (the server list appears as part of the component configuration) and none of the SMTP gateways are available and functioning. The OUT QUEUE directory is the "retry" directory for AutoStore server. If the server has not been able to contact outgoing SMTP gateways and send e-mail correctly, it will continue to try until all e-mail is sent.

Problem	Solution
The menus appear on the MFP device, but the correct menu items do not appear.	Several setup errors might create the incorrect menus: 1. The device is pointing to the wrong AutoStore server. Check the Servlet AutoStore server address and make sure that it is pointing to the correct AutoStore server. 2. The AutoStore server has more than one process that uses the MFP component as the Capture component. Only one process within AutoStore can include the MFP component as a Capture component. Creating multiple processes that use the MFP Capture component forces these components to compete for the same IP Port resources and device connectivity. 3. The device is a member of one of the device groups and it is taking the menus from the device group (and not from the common group). Make sure that the device appears in only one group. A single device that appears in multiple groups will cause the groups to compete for the same resources, and the groups will attempt to synchronize the device menus incorrectly.

Restrictions and limitations

- An MFP device can appear in only one device group.
- Another server process cannot use port 25. If your AutoStore server fails to start, check to see whether or not another server process is already using port 25; for example, the SMTP service. You can also use the Telnet program to gain access to port 25 to check whether or not it is in use.
- Only one AutoStore process can use the MFP (4100/9000) component as its Capture component.
- Use of the following characters is illegal within the **Form** or **Action** button names:
  - @
  - $
  - &
  - <
  - >
  - ?
  - /
  - \
  - |

LaserJet 9055/9065 MFP component

The LaserJet 9055/9065 MFP component communicates with the Chai .JAR software program files that are installed on the HP LaserJet 9055mfp and HP LaserJet 9065mfp devices. The MFP 9055/9065 component receives electronic images and information about the image for all of the HP LaserJet 9055mfp and HP LaserJet 9065mfp devices on which it is installed, and processes the information. Using the configuration parameters within the LaserJet 9055/9065 MFP component, the server captures the electronic file based on the function-key definitions.

Feature highlights

The LaserJet 9055/9065 MFP component offers the following features in the user interface (UI):

- **Scan Documents.** Go to the Scan/Server mode, select **Network**, press the button(s) that identify the destinations that you want, and select **OK**. You can add or change these by pressing **Address** from the Scanning screen. To finish scanning, follow the instructions on the scanning screen.
- **Create and manage the device button directly from the LaserJet 9055mfp or LaserJet 9065mfp Configuration tab on the LaserJet 9055/9065 Capture component in the AutoStore Process Designer (APD).** Use the buttons on this page to add, edit, and modify the button entries for each button.

Using the LaserJet 9055/9065 MFP component

You can use the LaserJet 9055/9065 MFP component to perform the following tasks:

- Capture files from MFP devices and route the files to a variety of destinations, such as Folder Store, FTP Store, or other components.
- Create buttons or function keys for forms on supported HP MFP devices.
- Create groups of HP MFP devices and maintain buttons or function keys that are based on the device group.

Configuring the LaserJet 9055/9065 MFP component

The AutoStore support for MFP devices provides an easy-to-use interface to create workflow connectivity and menu buttons, and to control all other related electronic-document processing attributes for MFP devices. To configure MFP devices on the server, click the **Configure** button (**...**) that appears in the last column of the Configuration page.

MFP Menu tab

Each **MFP Menu** tab contains at least one entry for each of the most common function-key definitions. This Common MFP Group appears at the top of the list. By default, any MFP that is not a member of another MFP group is considered to be a member of this group.

Use the Common MFP Group to create the default button definitions that appear on the most frequently used MFP devices. If new MFP devices are not added to any specific Device Group, then the Common MFP Group-definition schema is used.

- **Add Group.** Use the **Add Group** button to create unique LaserJet 9055mfp or LaserJet 9065mfp groups that are configured differently than the Common Group. Each group must have at least one device member, and each group requires the following information:
  - **Group Name.** A string value describing the name of the group. This name does not appear anywhere on the MFP device and it is used strictly as a logical group name.
  - **Group Devices.** Use the following buttons to manage the entries on this list:

    Add. Add a device to the list. You can use the device network name or the device IP address. It is recommended that you use a network name to accommodate possible changes in IP addresses when using the dynamic host configuration protocol (DHCP).

    Remove. Click a device entry and press **Remove** to remove a device from the list.
- **Add Button.** Add a button to capture index data and describe the processing attributes. To add a button, click the **Add Button** button and fill out the **General**, **HPS Generation**, and **Components** tabs.
- **Edit.** Edit an existing group menu or form by highlighting an object and then clicking the **Edit** button.
- **Remove.** Remove an existing group, menu, or form by highlighting an object and then clicking the **Remove** button. This operation cannot be undone.

General tab

Field name	**Description**
Scanning Mode	Select to scan a black-and-white document (1 bit per pixel, 400 x 400 DPI) or a fine-text OCR (1 bit per pixel, 600 x 600 DPI) document.
File Format	Select one of the following options from the drop-down list: ● MTIFF ● PDF
Button Name	The button name that appears as the last key on the form. Use a descriptive action name such as "Scan" or "Scan to" to indicate its function.

HPS Generation tab

The **HPS Generation** tab provides the basic attributes that control .HPS file generation. These attributes activate .HPS file generation, provide destination information, and control the file pass-through to other components within the same process.

Field name	Description
Generate HPS	This function can be set on or off. Activates the .HPS file generation.
Title	This string becomes the Title field within the .HPS file.
Destination folder	The folder path into which the .HPS folder and image file will be stored.
Pass-through	This function can be set on or off. **On:** The image file is processed as configured within the **Component** tab. **Off:** The process stops after generating the .HPS file and the image files. The files will not be processed by the remaining components within the process. Turn pass-through off when you need only an .HPS image file and you do not require files to be processed by remaining components.

Components tab

This tab lists all of the Process components with their description and type, and provides a **Configure** button so that you can set the attributes. Click a component name to select it, and then click the **Configure** button to configure its attributes.

Preferences tab

The **Preferences** tab provides the basic attributes that control how MFP messages are stored, forwarded, and routed. Type the file attributes and port number information on this tab.

- **Home Directory.** This is the root directory where AutoStore creates all temporary directories and files that are necessary for controlling the MFP-related traffic. The AutoStore server must have adequate permission to write to this directory.
- **Port Number.** This is the IP port number that is used between the MFP device and the AutoStore server for communicating the Chai servlet configuration attributes. This port number *must* match the port on the Chai servlet.
- **Keep.** To determine where your files are stored, select one of the following check boxes:
  - **Processed Files.** The Processed Files directory is where the captured files (send to folder or program files) are stored upon the completion of successful routing. If you turn this attribute on, all files are stored to the Processed Files directory when AutoStore successfully completes writing them to the destination location.

CAUTION If this is turned on, a copy of every file that has been successfully routed is saved to the Processed Files directory. Make sure that adequate disk space is allocated to this directory.

  - **Rejected Files.** The Rejected Files directory is used to store any files that failed to store to the final destination location. The failure could result from any number of reasons, and the failure must be researched by reviewing the program log file entries. Typically, HP recommends having a secondary process that takes this directory as an input and routes messages to the system administrator to warn about the failure. AutoStore must have sufficient rights to access this directory.

Using the LaserJet 9055/9065 MFP component in a multiprocess chain

A multiprocess chain is a set of AutoStore processes in which output from one process feeds into the input of another process. Multiprocess chains are useful when you have images or data elements that must be routed to multiple destinations.

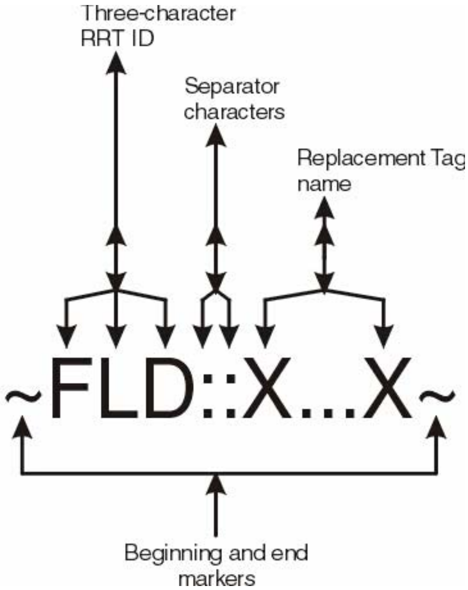
NOTE The LaserJet 9055/9065 component provides a **Success** or **Reject** option on the **Preferences** tab. All successful and rejected files are stored as .HPS+Image file pairs. You can use the Poll Directory component to read files from the Success directory by specifying the file type (.PDF or .TIF). In this way, the .HPS file is ignored. When Poll Directory is used in the second process within the chain, AutoStore does not process the .HPS file format portion of the files.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **DCS**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
ScannerName	The fully qualified host name of the sending device.
Sender	The Sender RRTN references the user that scanned the job if the “Send Monitor Function” of the LaserJet 9055/9065 is on.
IP	The IP address of the sending device.
ScannerModelName	The product name of the sending device.
Pages	The number of pages in the document.
Format	Format of the document, which can be one of the following values: ● Multipage .TIF file format ● .PDF file format
ApplicationItem	The name of the button that the user pressed to generate the job.

The following is an example of the RRTN process:

~DCS::ScannerName~(~DCS::IP~) is replaced with the value "LJ9055.mydomain.com (192.168.0.13)" if the fully qualified name of the device is LJ9055.mydomain.com and the IP address is 192.168.0.13.

Special set replacement tag name (SSRTN). The DCS component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale

SSRTN	Description
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"~DCS::%Y~-~DCS::%m~" will be replaced by "2003-9"

Troubleshooting tips

Problem description	Solution
The device does not show the menu buttons.	The device is not configured to point to an AutoStore server. Check the device configuration and make sure that the Chai servlet is pointing to the correct AutoStore server. If you change the configuration of the AutoStore server to use a different port rather than the default 3434, make sure that the Chai servlet is configured to connect to the appropriate port. Apply the changes to the configuration and you should be able to see the menus in the device after a few seconds.
The AutoStore server does not start. The log entry shows the following message: "Can't create directory."	The folder paths that appear on the **Preferences** tab must all exist. The AutoStore server attempts to create these folders, but if it is not successful or if the folder is deleted before the AutoStore server starts, then the server will not start correctly. Make sure that the folders on the **Preferences** tab exist, and that the AutoStore user has full access to them.

Problem description	Solution
The menus appear on the MFP device, but the correct menu items do not appear.	Several setup errors might create the incorrect menus: 1. The device is pointing to the wrong AutoStore server. Check the Servlet AutoStore server address and make sure that it is pointing to the correct AutoStore server. 2. The AutoStore server has more than one process that uses the MFP component as the Capture component. Only one process within AutoStore can include the MFP component as a Capture component. Creating multiple processes that use the MFP Capture component forces these components to compete for the same IP Port resources and device connectivity. 3. The device is a member of one of the device groups and it is taking the menus from the device group (and not from the common group). Make sure that the device appears in only one group. A single device that appears in multiple groups will cause the groups to compete for the same resources, and the groups will attempt to synchronize the device menus incorrectly.

Restrictions and limitations

Use of the following characters is illegal within the **Form** or **Action** button names:

- @
- $
- &
- <
- >
- ?
- /
- \
- |

Poll Directory component

Use the Poll Directory Capture component to batch load files into your programs. By using the Poll Directory component, you can import documents of any format from any shared network drive, process the content, and store the content in a supported document within the database management system.

The Poll Directory component is designed to poll an Input directory for files and then move all of the files from the Input directory to the Working directory before introducing them into the process. You can poll the Input directory for all files, or narrow your poll to a specific file type. Use wild-card strings to match certain file extensions or file names. (For example, `C:\AutoStore\Sample\Inbox\*.TIF` reads all files with the .TIF file extension into your process.) The Poll Directory component moves all of the files from the Inbox directory to the Working directory before introducing them into the process.

Feature highlights

The following items are the major features of this component:

- Poll Directory accepts any type of input file.
- Poll Directory moves files into the Working directory.
- Poll Directory can be a Capture component when paired with any Process or Route component.

Using Poll Directory

Use the Poll Directory component in the following ways within a process:

- Implement record migration between document management systems by exporting files into a directory and importing the files by using Poll Directory.
- Import scanned images from CDs into your back-end document management programs by using Poll directory.
- Read fax .TIF images by using the Poll Directory component, and then review them and route them into back-end programs. Poll Directory is the recommended component to connect your fax servers into back-end programs.
- Administrators can automatically capture documents from shared network drives without any change in the users' day-to-day process. You can continue to capture documents on the shared network drive, while, in the background, the process captures the documents in a document management system.
- Make the Success or Failure directory of one process feed into the inbox of the next process, and use another process to read the images. For example, the first process has a Digital Sender Capture component that uses the Success directory C:\PROCESS1\SUCCESSDIR. The next process in the chain uses a Poll Directory component and sets its Inbox directory to C:\PROCESS1\SUCCESSDIR. Notice that with this scenario, the files that are placed in Process 2 are the same files that were placed in Process 1. (The same files were fed to both processes in this chain.) Use this technique if the same files must be routed to different destinations.

Configuring the Poll Directory component

Configure the Input and Working directories for this component.

Input directory

You can poll the Input directory for all files, or narrow your poll to a specific file type. Use wild-card strings to match certain file extensions or file names.

For example, `C:\AutoStore\Sample\Inbox\*.TIF` reads all files that have the .TIF file extension into your process. You can also limit your file-matching parameter to a directory name and then read all files from that directory to the process. The poll directory reads files to a process one file at a time until the Input directory has no more matching files.

Working directory

This directory removes documents from the Input directory so that a document will not be picked up again. All documents in the Input directory are processed. Therefore, when you remove a document and place it into the Working directory, it prevents the infinite processing of the same document.

NOTE The files remain in the Working directory and are not automatically removed. If you want to remove the files from the Working directory, place a File Options Process component in your process and configure it to remove the files.

Using the Poll Directory component in a multiprocess chain

A multiprocess chain is a set of AutoStore processes in which output from one process feeds into the input of another process. Multiprocess chains are useful when you have images or data elements that must be routed to multiple destinations. The following cases illustrate how to use the Poll Directory Capture component in a multiprocess chain.

NOTE You must use the Poll Directory Capture component and the File Options Process component to start a multiprocess chain. (The File Options component is not included with the AutoStore Server software. It is an additional component that is available separately.) Poll Directory by itself does not have the **Success** or **Failure** option, but File Options does provide this option. To use a multiprocess chain based on the success or failure of file processing, use the Poll Directory and File Options components.

Case 1: You want to use the Poll Directory Capture component to read files, and then send them to a folder that you specified in the Route component. From the destination folder, you want to use the File Options component to send the files to a Success directory or a Failure Directory. You want the files in the Success directory stay in that destination folder, and the files in the Failure directory to go to a different folder.

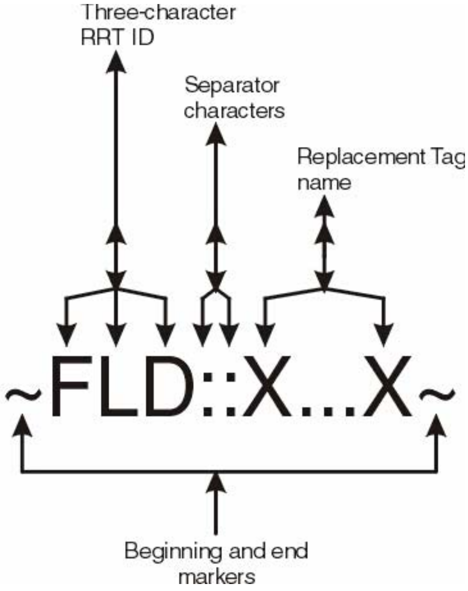
Case 1 solutions: After the first process sends the files to the specified folder, the second process in the chain uses the File Options component to capture the files into a Success or Failure directory. The files in the Success directory are through being processed. In the third process in the chain, the files in the Failure directory are captured by using the Poll Directory component, and processed by using the File Options component. The Route component sends the files to other folders, or sends failed files back to the Failure directory. You can use a Route component such as Send to Mail Recipient to notify the network administrator that some of the files failed during processing.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Poll Directory component does not generate RRTs.

Troubleshooting tips

Problem description	Solution
Errors appear in the log while accessing files in the Input directory.	● Check the user permission and make sure that the User ID has access to the Input and Working directories. ● Check the path name for the Input directory, and make sure that the Input folder is available to the server. ● Make sure that the server has full control over the path (including write and delete permissions) for the Input and Working directories.
Files remain in the Working directory when using Poll Directory with ABBYY FormReader.	When you are using the Poll Directory component with the ABBYY FormReader component, files are moved from an Input directory to a Working directory, where they remain. To remove these files and avoid using large amounts of disk space, add the File Options component to the HP AutoStore process and set the **Remove Files** attribute on the **General** tab.

Problem description	Solution
The HP AutoStore process containing the Poll Directory component does not function.	The Poll Directory component is not a mapping component. You must include the Data Interchange Process component to create a mapping component in the HP AutoStore process. The Data Interchange component provides the mapping capability from which the rest of the workflow is configured. For more information about mapping components, see Types of components.

Restrictions and limitations

- The Poll Directory reads one file at a time. For example, if your Inbox directory contains 200 files that match your polling criteria, the Poll Directory will introduce 200 separate files into your process.
- Each process can have only one Poll Directory component.
- The Poll Directory component must appear at the beginning of each process.

AutoCapture Server component

Use the AutoCapture Server component to extend the AutoStore capture capabilities to your computer. Use this component to capture all of the file types that reside on your computer into the AutoStore process. The AutoCapture client component must be configured on a user machine that requires the capability to capture files from the computer and place them into an AutoStore workflow process.

Features

You can perform the following tasks with the AutoCapture Server component.

- Select the group or users that will be presented with the AutoCapture feature.
- Define a logical group name for the group or users that will be using specific processes.
- Customize the forms by defining your own style sheet.
- Specify the file types that are processed through this component.
- Create a menu hierarchy.

Licensing

A specific number of AutoCapture licenses are available every time the server is started. These licenses are available to clients on a first come, first served basis. For example, if 12 licenses are available, then the first 12 clients who attempt to connect to the server are allowed to connect to the server.

Using the AutoCapture Server component

The AutoCapture Server component comprises a server software that runs on the AutoStore server and a client software that runs on your computer. The AutoCapture client communicates with the AutoCapture server to retrieve the menu and form information for presentation on your computer. When the server and client software are configured, you can select files by right-clicking the mouse button to open the AutoCapture menus and forms.

Use this component to capture any type of file from a user computer and process it into an AutoStore workflow. The user computer that is running the AutoCapture client communicates with the AutoCapture server over a configured port to retrieve group, menu, form, and action displays for the client computer.

You can select a form to fill out index data for a file. If you do not want indexed data, you can select an action that processes the files through the AutoStore Process components and Route component that are defined in the workflow process.

NOTE This section describes the AutoCapture server configuration. See the AutoCapture Client help section for information about configuring the client.

Installing the client software on each client computer

Before using the AutoCapture Server component, you must install the client software on each client computer. Copy the executable file of the AutoCapture Client to the user computer where you plan to run the AutoCapture Client software. The executable file is located inside the AutoStore default directory: C:\PROGRAM FILES\HEWLETT-PACKARD \HP AUTOSTORE\AUTOCAPTURE CLIENT\AUTOCAPTURE CLIENT SETUP.EXE. Double-click on the executable file and follow the instructions that appear on your monitor to complete the installation.

Configuring the AutoCapture Server component

The following attributes are available in the **AutoCapture Client** configuration dialog box.

Menu tab

Use this tab to add, edit, or remove a group, menu, form, or action.

- Click **Add**, and then select **Group** to add a user group. Each user-group entry requires the following information.

Field Name	Description
Name	Type the name of the group that you are creating. This is a string value that represents the name of the group. This name *does not* appear on the computer and is used only as the logical group name. This component is equipped with a default group called Common Group. You can add a Menu, Form, or Action to the Common Group. When the device is configured, the user can right click on the file that requires processing. The user will be presented with an AutoCapture icon labelled Send To. At this point, the user can select the configured menu, form, or action.
Root Menu Title	Type a menu title. This menu title *does* appear on the client computer. The menu title should represent the task to which it is associated.
Member Name	Click **Add** to enter valid groups or users that you want to be able to use this menu entry. Click **Remove** to delete a group or user.

- Click **Add**, and then select **Menu** to create a hierarchy. The menu entry guides the user to select the correct AutoStore process.

Field Name	Description
Menu	Type the name of the menu entry.

- Click **Add**, and then select **Form** when your AutoStore process requires you to capture index data. When you add a form, you must fill out the **General** and **Components** tabs. The **Components** and **General** tabs are discussed in detail at the end of the Configuring the AutoCapture Server component section.
- Click **Add**, and then select **Action**. You can directly assign an action to a menu entry item if the computer user has no need to assign index fields. For example, you can select a file that does not require any indexing field information to be captured and apply that file directly to the AutoStore process. Depending on the Process components and the Route component in that particular process, the file is processed and then stored in the designated Route component.

  For every action that you create, you are configuring a distinct AutoStore process that is based on the Process and Route components that you select. If the process requires index fields to be added to the stored component, you must create a form.
- Select the group, menu, form, or action that you want to update, and then click **Edit** to make the updates to your selection.
- Select the group, menu, form, or action that you want to delete, and then click **Remove** to delete your selection.

Preferences tab

Use this tab to set the AutoStore storage location for program files that the current job is processing.

- **Home Directory.** Specify the root directory where AutoStore creates all temporary directories and files that are necessary for controlling device-related traffic. The AutoStore server must have the appropriate authorization to write to this directory.
- **Port Number.** Type the port number that the AutoCapture client uses to communicate with the AutoCapture server. This port number must match the port number that is used when configuring the AutoCapture client computer.
- **Keep.** Keep the files in either a Processed File directory or a Rejected Files directory.
  - **Processed Files.** The directory where the captured files (send-to folder or program files) are stored after successful routing. If you turn on this attribute, all of the files are stored in the Processed Files directory when AutoStore successfully completes writing them into the destination location. Make sure that you have enough disk storage allocated for this directory.
  - **Rejected Files.** Use this directory to store any files that failed to store to the final destination location. The failure could be the result of several occurrences. The program-log file entries contain the failure information. Typically, having a secondary process that takes this directory as an input and routes a failure message to a system administrator is a beneficial approach.

    AutoStore must have the appropriate authorization to use this directory.

General tab

Use this tab to set the following attributes.

- **Form Name.** Type the name of the form that you want to appear on the computer hierarchy.
- **Button Name.** Type a button name that will appear at the lower-right corner of the form. Use a descriptive name, such as "Submit."
- **Style Sheet.** Type the name of your style sheet. If you leave this field blank, the default style sheet is enabled. A sample style sheet and XML schema are available on your onscreen Help.

  You can create a your own .XML style sheet to customize the format of your form by using different font colors and sizes, logos, and so on, as your business requires. When you right-click on a file and select a form, the AutoCapture client retrieves the form definition from the server in .XML format. The .XML-formatted file contains all of the field setting definitions for that form. The style sheet converts the .XML-formatted form and shows it in .HTML format.

  The custom .XML style sheet must provide the appropriate version of the ACForm element that is defined in the AutoCapture XML schema.

  If the custom .XML style sheet contains references to external files, then you can enter a list of file names that are separated by commas or semicolons. The files should be listed in the following order:

  - Style-sheet name
  - Image-file name
  - Java-scripts file name
- Form Fields. Add fields by clicking **N**, which represents "new field." When you have added the first row, you can tab through to add more fields. For each field, you can configure the field attributes.
  - **Field name.** This is a string type and can be composed of any alphanumeric characters. Special characters (~, &, <, >, %, ", /, \, :, ?, and |) are not recommended.
  - **Type.** Supported string types are Boolean, date and time, string, string list, and string multiline. The date and time types supported are listed in a table at the end of this section.
  - **Required checkbox.** Select this check box if you want the field to be required.
  - **Help.** The help message appears in the client as a tool-tip window when the curser is moved over a field on the form. The help message can serve as a guide when a user is filling out a form.
  - **Properties.** To set the field attributes for each entry, click the **Properties** button, which is located on the rightmost column of each entry.

The following table lists the date and time types that are supported in the **Type** field on the **General** tab.

Format	Description
%a	The abbreviated weekday name

Format	Description
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%C	Century number
%d	The day of the month as a decimal number (01 to 31)
%e	The day of the month (1 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%k	The hour (0 to 23)
%l	The hour (1 to 12)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%n	A newline character
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%P	The a.m. or p.m. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%s	The number of seconds since Epoch (since January 01 1970 00:00:00 UTC)
%t	A tab character
%U, %W, %V	The week number
%u	The weekday as a decimal number (1 to 7; Sunday is 1)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number
%%	A literal % character

Components tab

This tab lists all of the Process components and their description and type. Use the configuration button "**...**" to set the configuration attributes. The configuration button appears in the last column of each row.

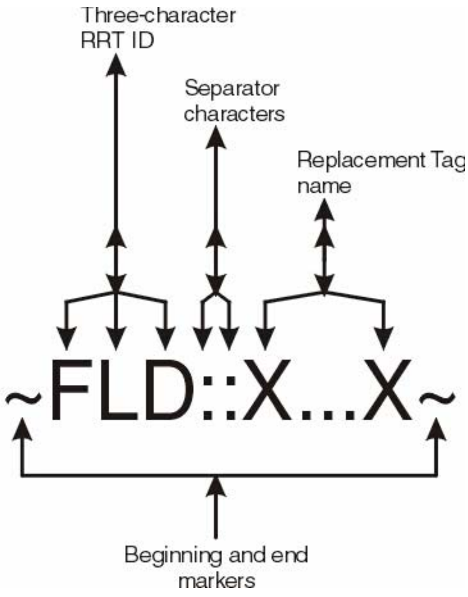
You can select a component by clicking the leftmost column next to the component that you want to select, and then click the configuration button to configure the attributes.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs)<xs:complexType name="ACField"> are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **ACC**.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
UserName	This is the log on user name of the client that is submitting documents.
Domain	This is the log on domain of the client that is submitting documents.
ComputerName	This is the NetBIOS name of the client's local computer that is submitting documents.

Field replacement tag name (FRTN). This component supports FRTNs and replacement of field names generated from the multifunctional device. The following is an example of an FRTN.

~ACC::%Client ID%~ is replaced by the value "Hewlett Packard" if you type `Hewlett Packard` in the field named, Client ID.

Special set replacement tag name (SSRTN). This component does not support any SSRTNs.

Restrictions and limitations

- Another process cannot lock the AutoCapture port (for example, 8085) cannot be locked by another process. If your AutoCapture fails to start, check for another process that could be listening on that port.
- Use of the <, >, &, or " character is invalid when used in the configuration of a **Form**, **Action**, **Menu**, or **Group** name.
- When you use Form Field names in the Rename Schema of a subsequent component, do not use the following characters:
  - /
  - \
  - :
  - ?
  - "
  - <
  - >
  - |

Troubleshooting

Troubleshooting information is not currently available for this component.

AutoCapture Client

Use the AutoCapture Client component to capture any file types that reside on the user's computer and place them into the AutoStore workflow process. The AutoCapture component extends the AutoStore capture capabilities to your computer.

You must configure the AutoCapture client component on the user computer that requires the capability to capture files from a computer and place them into an AutoStore workflow.

Using the AutoCapture Client component

The AutoCapture Client component comprises a server software that runs on the AutoStore server and a client software that runs on the user computer. The AutoCapture client communicates with the AutoCapture server to retrieve the menu and form information for presentation on the user computer. When the server and client software are configured, a user can select files by right-clicking the mouse button to open the AutoCapture menus and forms.

Configuring the AutoCapture Client

The following attributes are available in the **AutoCapture Configuration** dialog box.

- **Activate.** Select this check box to activate the AutoCapture functionality on the user computer.
- **Server.** Type the IP address or host name of the AutoCapture server.
- **Port.** Type the port number over which the AutoCapture client communicates with the AutoCapture server.

NOTE This port number must match the port number that is assigned to the AutoCapture server.

- **File Type(s).** Specify the types of files that will retrieve the AutoCapture menus from the server when a right mouse click is detected. A "." default enables the AutoCapture menus to accept all file types. To designate specific file types, use a comma or semicolon to separate a list of file-type extensions (for example, *.TIF, *.PDF, and so on).

Batch Importer component

Use the Batch Importer component to read and import ASCII-delimited files for processing. The imported index fields can also contain one or more file path names for importing associated image files. Use the Batch Importer component to define the delimiter, field orders, field names, and the image file field designation.

Feature highlights

Use the Batch Importer component to perform the following tasks:

- Import ASCII-delimited index files.
- Import one or multiple image files.
- Replace field values.

Using the Batch Importer component

The following examples represent the most common uses of the Batch Importer component.

- Import CDs that scanning centers generated. These types of CDs are usually ASCII-delimited text files that contain index information and a scanned-image file path. By using the Batch Importer component, these files and the corresponding index information can be imported into other programs.
- Import files such as form recognition files, check reading files, or other types of data files that software programs generated. The programs that generate these types of files can usually export the data into ASCII-delimited text files. By using the Batch Importer component, all of the data that these programs create can be used with AutoStore.
- Convert files from one program to another program by using batch text files. Use the Batch Importer component to migrate records from an existing system to another back-end program.
- Build custom interfaces. AutoStore offers VB scripting, Java scripting, VB, and VC++ so that you can build any type of custom program that uses the Batch Importer component.

Configuring the Batch Importer component

Use the appropriate procedure to open the **Batch Importer** configuration dialog box to configure the Batch Importer component.

Use static or dynamic values as defined in the Source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Batch Importer component.

The following attributes are available in the **Batch Importer** configuration dialog box.

Workspace tab

Use the options on this tab to set the following attributes:

- **File Type.** Specify the type of ASCII-delimited index files that will be polled from the Index directory (for example, *.CSV).
- **Input Directory.** Type the directory name of the directory that this component searches to find an index file. Type the directory name only (for example, `C:\SAMPLEINBOXDIR\`).
- **Working Directory.** Type the name of the directory that files move to after the process starts. Type the directory name only (for example, `C:\SAMPLEWORKDIR\`).
- **Processed Batch File.** After a batch file has been completely processed, the file can either be permanently removed or moved to a designated directory.
- **Rejected Record File.** Each row in the batch index file is considered a record that will be processed separately. If the processing fails on a specific record, the row of index data can be saved in a file and moved to a designated directory.

Field Setting tab

Use the options on this tab to set the following attributes:

- **Field Delimiter.** Select the character that separates the fields in the batch index file.
- **First Row Contains Field Names.** Select this check box if the batch index file contains column headings.
- **External File(s) Fields.**
  - **Contains External File.** Select this option if the index fields contain one or more file paths for importing.
  - **External File Fields.** Specify which index fields contain file paths. You can type either the Field Name(s) (column header), or the Field Number(s) (such as F1 or F2).
  - **External File(s) Must Exist.** Select this check box if you want the process to fail when the referenced external file path is invalid.
  - **On Success.** After the process has successfully completed, the external file(s) can either stay in its current directory, be permanently removed, or be moved to a designated folder.
  - **On Failure.** If the process failed to complete successfully, the external file(s) can either stay in its current directory, be permanently removed, or be moved to a designated folder.

Component Settings tab

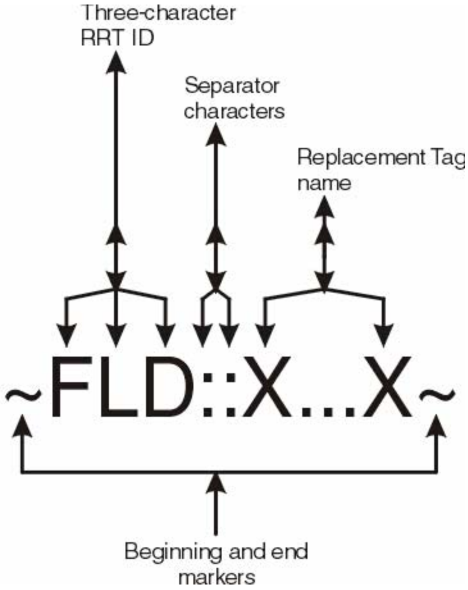
Use this tab to configure the components that follow the Batch Importer component in the process. You can configure any required field-value replacements by using the options on this tab.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **IBI**.

Use either the Field Name (**~IBI::Field1~**) or the Field Number (**~IBI::F1~**) to perform field value replacement.

Special Set Replacement Tag Names (SSRTN)

No SSRTN is supported by this component.

Troubleshooting tips

Problem description	Solution
AutoStore reports errors in the log while gaining access to the Input directory.	● Check the AutoStore user permission and make sure that the User ID has access to the Input and the Working directories. ● Check the path name for the Input directory and make sure that the Input folder path is available to the AutoStore server. ● Make sure that the AutoStore server has full control over the path (including write and delete permissions) for the Input and Working directories.

Restrictions and limitations

- This component reads one file at a time. When the entire batch index file is processed, the Batch Importer component reads the next file that matches the polling criteria in the Input directory.
- Each row in the batch index file is a separate record and is processed as a single job.

POP3 E-mail component

Use the POP3 E-mail Capture component to retrieve e-mail messages from a mail server that supports POP3 protocol. This component retrieves and processes the e-mail messages, including any attachments, from the designated mailbox one at a time. The content of each e-mail message, including any attachments, are retrieved and processed as a single job. Upon completion of the processing by the POP3 E-mail component, the e-mail message is removed from the designated mailbox.

POP3 stands for Post Office Protocol, Version 3. POP3 is a mail protocol that is used only to retrieve mail (from a POP3 enabled mail server), not to send mail. This component uses the POP3 protocol over transmission control protocol (TCP) to query a POP3 enabled mail server for new mail messages.

In order to use this component, you must supply a POP3 account name and its associated password for a specific mail. The POP3 account name is a unique name for an electronic mailbox. This information must be unique so that only the intended owner of a mailbox can gain access to it.

Feature highlights

You can perform the following tasks by using the POP3 E-mail component.

- Retrieve e-mail messages, including attachments, from a designated mailbox on a mail server that supports the POP3 protocol.
- Save e-mail messages as a file. The generated file is either a text file or an .HTML file.

Using the POP3 E-mail component

The following examples represent the most common uses of the POP3 E-mail component.

- Scan documents by using a local scanner and e-mail the scanned documents into a designated POP3 E-mail mailbox for processing.
- Forward incoming faxes to a designated mailbox for processing.
- Convert data to .PDF file format, and then maintain the data in an archived database. Use the POP3 E-mail component as the Capture component, the PDF Converter component as the Process component, followed by the Folder Store component as the Route component. In this scenario, the POP3 E-mail inbox reads the e-mail and converts the e-mail and attachments to .PDF file format, and then stores the data in a directory. Other users can archive documents by sending e-mail with attachments to the designated POP3 E-mail account. The process automatically reads the POP3 E-mail and converts the attachments to the .PDF file format and stores them in the archival system.

Configuring the POP3 E-mail component

Use the appropriate procedure to open the **POP3 E-mail** configuration dialog box to configure the POP3 E-mail component.

General tab

Use this tab to set the following attributes.

- **Server.** Type the name or IP address of the mail server from which the e-mail message will be retrieved. This mail server must support the POP3 protocol.
- **User Name.** Type the account user name that will be used to connect to the mail server. The POP3 account user name is a unique name for an electronic mailbox. This information must be unique so that only the intended owner of a mailbox can gain access to it. The POP3 account name is very often the first part of an e-mail address. For example, if an e-mail address is "**bobjones@xzy.com**", the POP3 account name is probably "**bobjones**".
- **Password.** Type the password for the user name that will be used to connect to the mail server.
- **Port Number.** The port number that the POP3 E-mail component will use to establish a TCP connection to the mail server. The default port for the POP3 connection is **110**.
- **Save email message as file.** Select this check box to save the e-mail header (that is, To, From, Subject, and so on) and the e-mail body as a text or .HTML file.

  This option does not affect the existing e-mail file attachments. You can create a new file that contains only the e-mail header and body content.

Component Settings tab

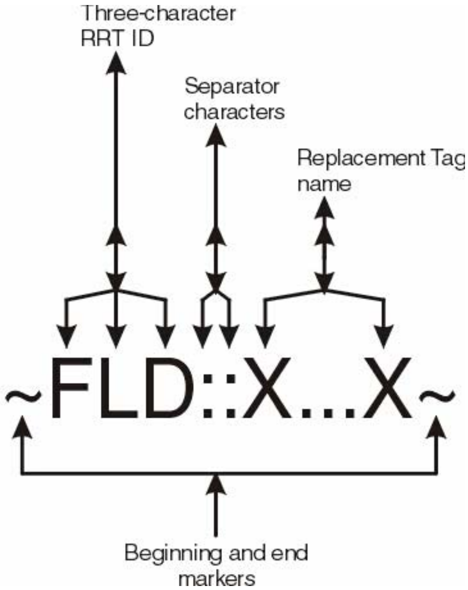
Click this tab to see a list of all process component names and their descriptions. Highlight the component that you want, and then click **Configure** to configure all of the attributes for that component.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **POP**.

The following RRTs can be used in any of the components that follow the POP3 E-mail component in an AutoStore process.

For example, if the SharePoint Portal Process component follows the POP3 E-mail Capture component in an AutoStore process, you can define the **Subject** field for the SharePoint Process component as **~POP::Subject~**.

- **~POP::From~**
- **~POP::To~**
- **~POP::CC~**
- **~POP::Subject~**
- **~POP::Date~**
- **~POP::Body~**

Troubleshooting tips

Problem description	Solution
The Status monitor indicates that the component is unable to connect to the POP3 E-mail server.	● Make sure that the POP3 protocol is enabled on the mail server. ● Make sure that POP3 support is not disabled for the user's mailbox. ● Make sure that the client can ping the server using the mail server Internet protocol (IP) address and the mail server computer name. ● Try using an e-mail client such as Outlook Express to connect to a user's mailbox. ● Use the instructions that follow to verify the connection to the POP3 mail server.

Problem description	Solution
The Status monitor indicates that the component is unable to connect to the POP3 E-mail server. (Continued)	1. Open a Telnet tool such as HyperTerminal. For the host name, type the name of the mail server computer. For the port type, type `110`. If a term type is requested, do not type anything. The following message appears: **telnet server 110** 2. Press Enter. The following message appears: **+ OK Microsoft Exchange 2000 POP3 server version 6.0.6249.0 ready** 3. Enable **local echo** and **Send line ends with line feed for the current telnet session**. 4. Type `user domain\username\mailbox`, and then press Enter. `domain` is the name of the domain in which the user's account is located, `username` is the user name, and `mailbox` is the user's mailbox. **NOTE** The mailbox portion of this command is necessary only if the mailbox is different from the username portion. 5. Type `pass password`, and then press Enter. `password` is the user's password. The following message appears: **+OK User successfully logged on** 6. To determine if the user has new messages, type `stat` and then press Enter. The following message appears: **+OK xyyyy** **x** is the number of new messages. **yyyy** is the total size of the messages in bytes. This is known as a "drop listing." 7. To end the Telnet session, type `quit`, and then press Enter. The following message appears: **+ OK Microsoft Exchange 2000 POP3 server version 6.0.6249.0 signing off**

Restrictions and limitations

You cannot gain access to messages from another e-mail client after the messages are downloaded to a local hard disk.

MFP/Digital Sender component

Use the MFP/Digital Sender component component to directly integrate with devices that are compatible with the HP LaserJet 4345mfp, HP LaserJet 9050mfp and HP LaserJet 9040mfp, HP LaserJet Color 9500mfp, and HP 9200C Digital Sender (HP MFP devices). Use this component to capture metadata from the device control panel and dynamically show data directly from your applications. Use the MFP/Digital Sender component component to integrate with HP Web Jetadmin for faster and more efficient configuration of your applications, along with faster and more efficient Chai loading capabilities.

Features

You can perform the following tasks with the MFP/Digital Sender component component.

- Create and manage groups of devices.
- Create and manage multiple menu levels for groups of devices.
- Create and manage forms that you can configure to capture metadata.
- Create and manage **Action** buttons to start a workflow process.
- Configure scan setting parameters to match the content that you want to capture at the device.
- Configure and manage the MFP control panel label text and image that appears. This is referred to as the application name.
- Securely communicate between the MFP and the AutoStore server by using Secure Sockets Layer (SSL).

Using the MFP/Digital Sender component component

Use the MFP9500/9050 component to design an AutoStore process that configures the workflow and capture behavior at the HP MFP devices.

Define a workflow process that stores a document in a network shared folder that corresponds to the MFP device user name and that also sends a notification message after the document has been successfully processed. For example, if the user "jdoe" with e-mail address jdoe@company.com scanned a document, this document is stored in \\share_name \jdoe, and a notification is sent to jdoe@company.com.

In this workflow, activating the **Credentials Required** option in the **Application** for this form requires the device to have authentication configured to facilitate a login that establishes user credentials. The user credentials that are collected at the device become available to the server for generating the e-mail and folder as described in the previous paragraph.

Configuring the MFP/Digital Sender component

The following attributes are available in the **MFP/Digital Sender** configuration dialog box.

- **General tab.** Construct groups, menus, actions, and applications. You can also use this tab to configure routing attributes.
- **Preferences tab.** Configure the settings for the AutoStore process home directory, IP port number, and other administrative directories.

General tab

Use this tab to configure group, applications, menu, form, and action attributes.

- Click **Add**, and then select **Group** to create additional HP MFP groups. HP MFPs that are members of specific groups have a menu structure that is independent of the Common MFP Group. You cannot define a group within another group.

  When you click the **Add Group** attribute, the **Group** configuration dialog box appears. Each group entry requires the following configured attributes.

Field name	**Description**
Name	Type the name of the group that you are creating. This is a string value that represents the name of the group. This name *does not* appear on the computer and is used only as the logical group name.
Members	Click **Add** to enter either a valid device IP address or device DNS host name that you want to define in the group. **NOTE** Using the network name is preferable because an IP address can change if you are using dynamic host configuration protocol (DHCP). Click **Remove** to delete a device from the group.

- Click **Add**, and then select **Application** to define a workflow. When an application is configured, the **Application** button appears on the MFP control-panel display. Define the application for a group of devices at the root level. A maximum of twelve **Application** buttons are supported. You cannot define an application within an application. You can configure an action, menu, and form within an application.

Field name	**Description**
Name	Type the name of the **Application** button. This is the label that appears on the MFP control-panel display. Any user can select this application to begin the corresponding AutoStore workflow process.

Field name	**Description**
Image	This is the image that appears on the MFP control-panel display before you press the **Application** button on the control panel. Click the browse button to locate the image that you want. This is not a required field. A default image is provided if you do not select one. The application supports .GIF images with five colors. The image size should be 106 pixels x 109 pixels. The recommended file size is up to 5KB. You can use the following RGB colors. Black: 0,0,0 Dark Grey: 131, 131, 131 Light Grey: 204, 204, 204 Transparent: 255, 255, 0 White: 255, 255, 255
Depressed Image	This is the image that appears on the MFP control-panel display after you press the **Application** button on the device control panel. Click the browse button to locate the image that you want. This is not a required field. A default image is provided if you do not select one. The application supports .GIF images with five colors. The image size should be 106 pixels x 109 pixels. The recommended file size is up to 5KB. You can use the following RGB colors. Black: 0,0,0 Dark Grey: 131, 131, 131 Light Grey: 204, 204, 204 Transparent: 255, 255, 0 White: 255, 255, 255

Field name	Description
Disabled Image	This is the image that appears on the MFP control-panel display when the AutoStore server is down or unavailable. Click the browse button to locate the image that you want. This is not a required field. A default image is provided if you do not select one. The application supports .GIF images with five colors. The image size should be 106 pixels x 109 pixels. The recommended file size is up to 5KB. You can use the following RGB colors. Black: 0,0,0 Dark Grey: 131, 131, 131 Light Grey: 204, 204, 204 Transparent: 255, 255, 0 White: 255, 255, 255
Credential Required	When this box is checked, the application requires the device to have authentication configured to facilitate a login to establish user credentials.
Application allows one touch scanning	Check this box to use the application as a standalone scan button. One-touch applications do not display any sub menus or forms, but they are used to start scanning immediately after the application button is pressed. This option is available only if the current application does not have any sub menus, forms, or actions associated with it. Check the one-touch scanning box and configure the scan settings by pressing the **Configuration** button. Note that the **Action** configuration window appears with the **Name** field disabled and displaying the application name.

- Click **Add**, and then select **Menu** to add a menu to create a hierarchy. The menu entry acts as a guide to help users select the correct functions.

Field name	Description
Menu	Type the name of the menu entry.

- Click **Add**, and then select **Form** when your AutoStore process requires you to capture index data. When you add a form, you must fill out the **General** and **Components** tabs. These tabs are described later in this section.

- Click **Add**, and then select **Action**. You can directly assign an action to a menu entry item if the computer user has no need to assign index fields. For example, you can select a file that does not require any indexing field information to be captured and apply that file directly to the AutoStore process. Depending on the Process components and the Route component in that particular process, the file is processed and then stored in the designated Route component.

  The **Action** dialog box contains two tabs, **Action** and **Component**.

  The following table shows the attributes available in the **Action** tab.

Field name	Description
Name	Type the name that is assigned to the action. The name that you assign to this field is the name that appears as the button name on the MFP control-panel display. Use a descriptive action name such as "Scan" or "Scan to" to indicate which procedure the **Action** button is going to activate.
Scan Setting	Use the drop-down list to configure the document scan settings: ● Page content (text, photo, mixed) ● File type (.PDF, .TIFF, .MTIFF, .JPEG) ● Resolution (75, 150, 200, 300)

  If the workflow process you are designing requires you to capture index data, then you need to create and design a form.

  See the **Components** tab section for a description of the attributes contained on this tab.

- Select the group, application, menu, form, or action that you want to update, and then click **Edit** to make the updates to your selection.

- Select the group, application, menu, form, or action that you want to delete, and then click **Remove** to delete your selection.

General tab (adding a form)

Use this tab to set the following attributes.

- **Form Name.** Type the name of the form that you want to appear on the MFP control-panel display. The maximum number of characters supported for this field is 25.

- **Scan Button Name.** Type a button name that will appear at the lower-right corner of the form. Use a descriptive name, such as "Submit." The default value for this field is "Send." The maximum number of characters supported for this field is 10.

- **Scan Setting.** Select from the following scan settings.
  - Page content (text, photo, mixed)
  - File type (.PDF, .TIFF, .MTIFF, .JPEG)
  - Resolution (75, 150, 200, 300)
  - Color (color, BW)
  - Multiscan check box (Select this check box to prompt the user after each scan job for additional pages to be scanned. This is a useful option when the documents are not contiguous, or when documents require the use of both the automatic document feeder (ADF) and the flatbed scanner.)
- **Form Fields.** Add fields by clicking **N**, which represents "new field," or click **Add**. When you have added the first row, you can tab through to add more fields. For each field, you can configure the field attributes.
  - **Field name.** This is the name of the prompt, which can be composed of any alphanumeric characters. Special characters (for example, ~, !, @, #, $, &, %, and so on) are not recommended. The maximum number of characters characters supported for this field is 15.
  - **Type.** Specify the type of data to use. Supported types are Boolean, date and time, string, string list, integer, float, and currency. These types are described in detail in a table at the end of this section.
  - **Required.** Select this check box if you want the field to be required.
  - **Help Message.** This help message is a short text message that appears on the device control-panel display and provides information to the device user. Click **?** that is located on the upper-left corner of the device control-panel display. When the help menu appears, select **What is this?**. Select the item that for which you are seeking help, you will see a message tip related to this item.

The following table describes the supported field types from the **Type** field that was defined previously. Use these field types to specify the type of data to use in the form fields.

Type name	Description
Boolean	Default Value: This is the default value that will be selected for this field. True Value: You can use a different value from True to identify a true case, such as Yes or On. If you do not specify anything here, True is the default. False Value: You can use a different value than False to identify a false case, such as No or Off. If you do not specify anything here, False is the default.

Type name	**Description**
Date/Time	Three modes are available for the this field. Format: Specify the desired format for this field. A valid format is composed of the following pattern letters. Date: With this mode only the day, month and year of the date can be specified. Time: With this mode only the hour, minutes, and seconds of the date can be specified. Date and Time: All of the date can be specified. The format for the Date/Time field is described in a separate table following this table.
String	Click **P** (Field Form Properties) to open the **String Properties** dialog box. This dialog box contains the following elements: Width in Characters - This control defines the maximum number of characters that are accepted. Default Value - This value is the default shown to the user on the MFP control-panel display. Users either accept the default or replace the default with a new value. Obscure Entry - Select this check box if you want to hide the input value that the user enters for this field.
String list	The String list creates a drop-down menu on the MFP control panel that is visible to users. However, users cannot change the values in the list. Click **P** (Field Form Properties) button when **String list** is selected in the **Type** field, to open the **String List Properties** dialog box. The String List Properties dialog box contains the following elements: N - Click the N column label to add a value to the list. Value - Use this field to type in the values a user sees in the drop-down menu. D - Select the check box in the D column to designate the corresponding Value as the default. Add - Click this button to add a new value to the list. Remove - Click this button to remove a value from the list.

Type name	Description
Integer	Click **P** (field form Properties) button when **Integer** is selected in the **Type** field, opens the **Integer Properties** dialog box. The **Integer Properties** dialog box contains the following elements: Min Value - Use this field to define the low end of the range. Max Value - Use this field to define the high end of the range. Default Value - This field defines the default value for the field. **NOTE** The user must specify a whole number value when using the integer field.
Float	Select **Float** type and click **P** (Field Form Properties) to open the **Float Properties** dialog box. The **Float Properties** dialog box contains the following elements: Precision - This field indicates the number of digits to the right of the decimal point. Min Value - Use this field to define the low end of the range. Max Value - Use this field to define the high end of the range. The Default Value is the number displayed initially. Default Value - This field defines the default value for the field.

Type name	Description
Currency	Select **Currency** type and click **P** (Field Form Properties) to open the **Currency Properties** dialog box. The **Currency Properties** dialog box contains the following elements: Min Value - Use this field to define the low end of the range. Max Value - Use this field to define the high end of the range. The Default Value is the number displayed initially. Default Value - This field defines the default value for the field. Precision - This field indicates the number of digits to the right of the decimal point. Currency Format - The user has two options to select from: pre-defined formats or custom formats. If the user selects a pre-defined format, the currency is displayed according to the locale selected from the list. Currently the following currency locales are supported: - Danish - Dutch (Netherlands) - Finnish - French (France) - German (Germany) - Italian (Italy) - Japanese (Japan) - Norwegian - Portuguese (Portugal) - Spanish (Spain) - Swedish (Sweden) If the user selects a customized format, the following parameters must be configured: - Currency Symbol: Specifies the symbol that identifies the currency of a country/region. - Positive currency format: Specifies how positive currency values are displayed. - Negative currency format: Specifies how negative currency values are displayed. - Decimal Symbol: Specifies the symbol that separates units of currency. - Digit grouping symbol: Specifies the symbol that groups the digits in large currency values. - Digit Grouping: Specifies the number of digits that appear between digit grouping symbols.

The following table describes the Date/Time field formats. Create a valid format with the following pattern of letters.

NOTE All lowercase and uppercase characters that are not described in this table are reserved.

Letter	Date or time	Presentation	Example
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	AM/PM marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC time zone	-0800

Letters that do not have quotations marks, from 'A' to 'Z' and from 'a' to 'z' are interpreted as pattern letters that represent the components of a date or time string. Text can be quoted using single quotes (') to avoid interpretation. A single quotation mark is represented by " ' ". All other characters are not interpreted; they are copied into display date.

Pattern letters are usually repeated because their number determines the exact final display, depending on the type of presentation:

- Text: If the number of pattern letters is four or more, the full form is used; otherwise a short or abbreviated form is used if available.
- Number: The number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount

- Year: If the number of pattern letters is two, the year is truncated to two digits; otherwise it is interpreted as a number.
- Month: If the number of pattern letters is three or more, the month is interpreted as text; otherwise, it is interpreted as a number.
- General time zone: Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:

  GMTOffsetTimeZone:

  GMT Sign Hours : Minutes

  Sign: one of

  + -

  Hours:

  Digit

  Digit Digit

  Minutes:

  Digit Digit

  Digit: one of

  0 1 2 3 4 5 6 7 8 9

  The hours must be between 0 and 23, and minutes must be between 00 and 59.

  RFC 822 time zone: For formatting, the RFC 822 4-digit time zone format is used.

  RFC822TimeZone:

  Sign TwoDigitHours Minutes

  TwoDigitHours:

  Digit Digit

  TwoDigitHours must be between 00 and 23.

  Other definitions are the same as the general time zone definition.

The following table provides examples of how date and time patterns are interpreted if the device is set to English. The given date and time are 2001-07-04, 12:08:56 local time in the United States PCT zone.

Date and time pattern	**Result**
yyyy.MM.dd G 'at' HH:mm:ss z	2001.07.04 AD at 12:08:56 PDT
EEE, MMM d, ''yy	Wed, Jul 4, '01
h:mm a	12:08 PM
hh 'o''clock' a, zzzz	12 o'clock PM, Pacific Daylight Time
K:mm a, z	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM

Date and time pattern	Result
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700

Components tab

This tab lists all of the process components and their description and type. Use the configuration button "**...**" to set the configuration attributes. The configuration button appears in the last column of each row.

You can select a component by clicking the leftmost column next to the component that you want to select, and then click the configuration button to configure the attributes.

Editing and removing

Under the main **General** tab, select the group, application, menu, form, or action that you want to update, and then click **Edit** to make changes.

Under the main **General** tab, select the group, application, menu, form, or action that you want to delete, and then click **Remove** to delete it.

Preferences tab

Use this tab to control the MFP internal attributes for storing, forwarding, and routing MFP messages.

- **Home Directory.** Specify the root directory where AutoStore creates all temporary directories and files that are necessary for controlling device-related traffic. The AutoStore server must have the appropriate authorization to write to this directory.
- **Port Number.** Type the port number that the MFP/Digital Sender component uses to communicate with the MFP device. The MFP/Digital Sender component listens to this port number to receive requests for menus, receive scan jobs, and to send menu updates to the MFP devices. The default value for the port number is 3233. This port number must match the port number that is used when configuring the MFP device.
- **Keep.** Keep the files in either a Processed File directory or a Rejected Files directory.
  - **Processed Files.** The directory where the captured files are stored after successful routing. If you turn on this attribute, two files are stored into the Processed Files directory. The first file is an .XML-formatted file that contains metadata and other internal data related to the form and action (if any exists). The second file contains the image.

    If this attribute is enabled, a copy of every file that has been routed successfully is saved in the Processed Files directory. Make sure that you have enough disk storage allocated for this directory.
  - **Rejected Files.** Use this directory to store any files that failed to store to the final destination location. The failure could be the result of several occurrences. The program-log file entries contain the failure information. Typically, having a secondary process that takes this directory as an input and routes a failure message to a system administrator is a beneficial approach.

    AutoStore must have the appropriate authorization to use this directory.

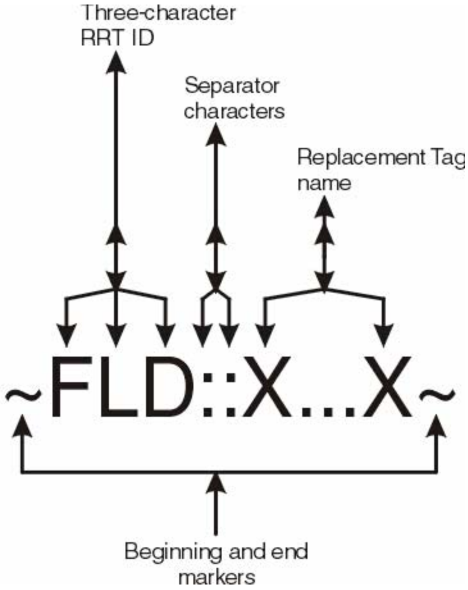
- **Job Transfer.** Use the **Maximum number of transfer retries** attribute to specify the number of times that the device will try to send a failed job. If a job fails for any reason, the device will try to resend the files when communication is re-established with the AutoStore server. Note that the **Interval** attribute in the MFP configuration dictates the time interval between the attempts made by the MFP that is trying to connect to the AutoStore server.
- **Connection.** Activate the **Use SSL for all connections** attribute to enable the information exchange between the AutoStore server and the MFP device over a secure channel.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **HPM**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
PageCount	This is the number of pages that are received.
Format	This is the file format value. ● .PDF: 1 ● .MTIFF: 2 ● .TIFF: 3 ● .JPEG: 4
FileSize	This is the size of the file that is received.
IP	This is IP address of the MFP device.
HostName	If available, this is the hostname of the MFP. If this is not available, it is the IP address.
ModelName	This is the model name of the MFP. For example, MFP9500, MFP9050, MFP9200, MFP4345, or MFP9040.

The following is an example of the RRTN process:

~HPM::PageCount~ is replaced with the value "10" for a ten-page document.

~HPM::Format~ is replaced with the value "4" if the file sent by an MFP is in .JPEG file format.

Field Replacement Tag Names

This component supports Field Replacement Tag Names (FRTNs) and replacement of field names that the MFPs generate. The following is an example of an FRTN:

~HPM::%Client ID%~ is replaced by "Hewlett-Packard" if someone types `Hewlett-Packard` for the field name "Client ID".

If authentication has been turned on for the MFP, then **~HPM::%Sender%~** contains the domain\username value for the authenticated user. **~HPM::%SenderAddress%~** contains the e-mail address for the authenticated user.

Special set replacement tag name (SSRTN)

The MFP/Digital Sender component supports the Date/Time field names that appear in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

Troubleshooting tips

Problem description	Solution
The device does not show the Applications	The device is not correctly configured to point to the AutoStore server. Check the device configuration for HP AutoStore and make sure that the device is pointing to the correct AutoStore server. If you change the configuration of the AutoStore server to use a port that is different from the default port, make sure that the device is configured to connect to the appropriate port. Apply the changes that you made to the configuration. You should be able to see the menus in the device after a few seconds.
The server fails to start. The log entry indicates "can't create directory."	Check for the following: The folder paths that appear under the **Preferences** tab must all exist. The AutoStore server attempts to create these folders, but if it is not successful, or if a folder is deleted prior to starting the AutoStore server, then the server does not start correctly. Make sure that the folders on the **Preferences** tab exist and that the AutoStore user ID has full access to them.
The MFP device menus appear, but the menu items are incorrect. Where are the menu items coming from?	There are several scenarios that could cause the incorrect menu items to appear. ● The device is pointing to the wrong AutoStore server. Check the AutoStore server address and make sure that it is pointing to the intended AutoStore server. ● The device is a member of one of the device groups and it is taking the menus from the device group (and not from the common group). Also make sure that the device only appears in one group. A single device appearing on multiple groups will cause the groups to compete against the same resources and attempt to synchronize the device menus incorrectly.

Restrictions and limitations

- The MFP port (for example, 3233) cannot be locked by another process. If this component fails to start, check for another process that could be listening on that port.
- Use of the following characters is illegal within the **Menu**, **Form**, or **Action** button names:
  - &
  - <
  - >
  - "
- Changing workflow configuration should occur only outside of normal working hours (for example, late at night) when no users are using HP AutoStore workflow processes. Administrators should only make content changes after normal working hours as well. This prevents synchronization problems such as the MFP device entering a 49.4208 state.

6 Process components

AutoStore uses a Process component to manipulate the data stream (images or data elements) within a process. The Process components must always appear between the Capture component and the Route component. Each Process component can accept a defined input data type and produce a defined output type. Process components are optional within an AutoStore process.

NOTE A Process component operates only on its defined file format or data type. If the input does not match the supported input specification, then the Process component simply passes the data on to the next component without any manipulation.

Add a Process component after a Capture and before a Source component within an AutoStore process. The Process components are completed in the order that they appear in the process chain. The output of each Process component (file or data elements) is made available to the components that follow.

General types of Process components

Process components perform different types of operations. The following categories are available.

- **Extractor type.** The images, files, or other types of objects that pass through a process contain data elements in the form of imaged data (bar codes), fields (PDF-embedded fields), and content (OCR text, forms, and so on). All extracted data fields (for example, barcode values) are made available to the components that follow a Process component within an AutoStore process.
- **Enterprise connector type.** A Process component is an enterprise connector if it can store information in a destination and produce a link to the newly stored information. This type of Process component is useful when you want to store information in one central location and create links to other related systems (workflow automation, ERP, and so on). An example is the Microsoft SharePoint Portal server enterprise component. This Process component stores documents in the SharePoint Portal and can create URL links that point from other destinations back to a document.
- **Manipulator type.** This type of Process component manipulates the data content of the data stream and actually changes the data type, content, or format. This Process component type can split an image to multiple images, or despeckle the image. Both of these operations change the actual data content.

Send to Printer component

Use the Send to Printer component to send scanned images directly to a printer. You can also use this component to select the paper size, source, and zoom level. All print drivers for each printer must be installed and configured on the server.

Use the Send to Printer component to define and designate a printer as the destination for images that are sent to the server. This component enables the MFPs to define copying function keys that scan documents and automatically send them to a defined printer location.

Use the Send to Printer component to batch-print commonly used image-file formats such as .TIF and .JPG. You can also use the system to print in .PDF file format when you acquire the appropriate license.

Use the Send to Printer component to print documents to any printer that is configured on the computer where the system is running. The system can also switch between printers based on the name or the extension of the document filename.

The Send to Printer component can be used both as a Process component and a Route component in a process workflow, depending on your needs.

The Send to Printer component manipulates different file formats without using any native program in the service computer. Because the printing process uses the printer drivers that are installed in the service computer, you must set up before configuring any printers that you want the component to use.

Feature highlights

Many business processes that use image documents incorporate printing as part of their workflow. The Send to Printer component satisfies many of the printing needs in a document-management environment. Whether obtaining a printed version of a document is the final goal of the process, or the system relies on printed versions as an alternate source of information, the Send to Printer component can be used to accommodate many scenarios where unattended printing is necessary.

Use the Send to Printer component to specify parameters that are common to most print drivers, such as paper size, printer name, pages per sheet, and orientation. Other parameters that are specific to a printer must be set in the local printer configuration.

The Send to Printer component has restricted capabilities for the type of files that can be printed. Only non-interactive printer drivers are supported. Some printer drivers might not be supported when the system is running as a service, because some printer drivers require an interactive response from you before printing.

Using the Send to Printer component

To use the Send to Printer component, you must first decide whether sending to a printer is the final step of the workflow or an intermediate step. This defines whether Send to Printer is used in the configuration as a Process component or as a Route component.

Use the Send to Printer component to accomplish the following tasks:

- **Remote copy.** Scan files and route them to a remote printer.
- **File-type based printing.** Scan files and send them to printers according to file types.

- **Color copy.** Scan in color and send the images to a color printer.
- **Print broadcast.** Create process chains with the Poll Directory Capture component and the Send to Printer Route component to broadcast multiple copies of a document to multiple printers.
- **Special printer features.** Accomplish automatic stapling and other specific printer functions by creating a copy of the printer driver that has the option for the function activated by default, and then using that specific printer driver.
- **Printing confirmation page.** Use File Options within your workflow process to store documents into a Success or Failure directory. If you want a confirmation page to report success or failure, use the Poll directory with the Send to Printer component to route files from the Success or Failure directory to a printer.

Licensing the Send to Printer component

The following types of licensing are available for this component:

- **Image Files.** The Send-to-Printer supports all of the file formats described in the Restriction and Limitations section, except for .PDF file formats.
- **Image and .PDF Files.** The Send-to-Printer supports all of the file formats described in the Restrictions and Limitations section, including .PDF file formats.

NOTE In the License Manager, you must turn on the PDF Enhanced level of licensing in addition to the base-level license if you want the AutoStore software to support .PDF file formats.

Configuring the Send to Printer component

Use the Send to Printer component to route images to printers that are located anywhere on the network. This component provides remote copying capabilities.

Each page (tab) in the Send to Printer component represents a printer that can receive image files. The **General** tab defines the default printer, which is used for general-purpose print jobs. Based on the file extension, you can further determine the routing. For example, you can add a tab and define a separate printer for .TIF images, or define another printer for .PDF files.

The following attributes are available in the **Send to Printer Configuration** dialog box:

- **Activate.** This activates the Send to Printer component to send images to the printer.
- **Keep.** Select this check box if you want to keep the original file (and not delete it).
- **Printer Name.** Use this drop-down list box to select the MFP.
- **Paper.** Use this drop-down list box to select the size of the paper. The values that appear change in relation to the printer and print driver programs. **Paper Source** indicates the paper source based on paper sources that are available on the printer driver. Select the **Auto Select** option if you want the printer to use a tray that supports the paper size that you selected.
- **Number of copies.** Type the number of copies that you want to print. To print a complete copy of the document before the first page of the next copy is printed, select the **Collate** check box.
- **Orientation.** Use these options to select the print-media orientation for printing.

- **Zoom.** Use this drop-down list box to set the number of pages that appear on each printed page.
- **Add.** Use this button to add a printer tab for a defined file extension. All files that match the file extension are routed to the printer that is defined on this tab rather than to the **General** tab.
- **Remove.** Use this button to remove a tab (other than the **General** tab). You must select a tab and then click **Remove** to delete the selected tab.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **Send to Printer Configuration** dialog box and configure the Send to Printer component.

Using Knowledge Package Loader to configure the Send to Printer component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Printer component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Printer component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Printer component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Printer component

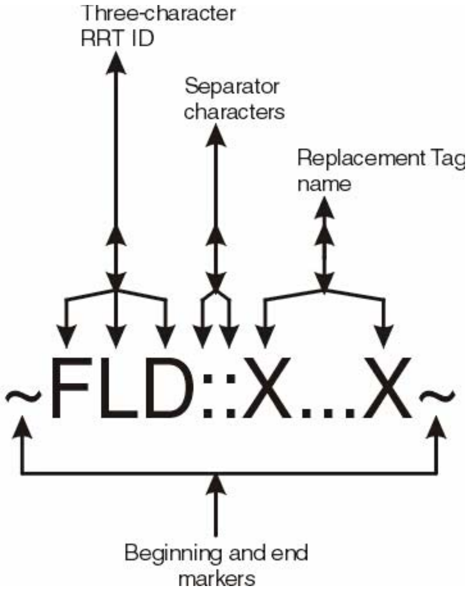
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to Printer component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Send to Printer component does not generate RRTs. However, all of the attributes can contain RRTs.

Troubleshooting tips

Problem	Solution
The server stops processing when you are trying to print document.	The print driver is trying to get interactive information. Make sure that the printer configuration does not request any information from you. For example, if the printer keeps a copy of the printing job, make sure that the file name is generated by the printer driver, rather than being requested from you. If you are using the system as a service, any kind of interactive actions cause the service to stop until a response is given. **NOTE** When you are using the interactive version of the system, the printer might need to request additional information from you.
An error occurs when you are trying to print a .PDF file, even though the system is licensed for .PDF file printing.	The .PDF file format that you are trying to use is not supported. If the program that generates the .PDF document can generate one of the supported .PDF file formats, use this instead (if your business process accepts the format).

Restrictions and limitations

Only non-interactive printer drivers are supported.

The following file formats are supported for Send to Printer.

JPEG formats

- JPEG File Interchange Format.
- Tagged Image File with JPEG compression.
- JPEG 2000 Format. This file format contains image data and extra information about the contents and organization of the file.

GIF formats

- CompuServe GIF.

TIFF formats

- Tagged Image File Format, with no compression and with RGB color space and 8-bit grayscale.
- Tagged Image File, with no compression and with CMYK color space.
- Tagged Image File, with no compression and with YCbCr color space.
- Tagged Image File with PackBits Compression and RGB color space.
- Tagged Image File with PackBits Compression and CMYK color space.
- Tagged Image File with PackBits Compression and color YCbCr space.
- Tagged Image File with CMP Compression.
- Tagged Image File with JBIG Compression.
- Tagged Image File with a vector image saved as a DXF file.
- Tagged Image File with JPEG 2000 Compression. This file format contains only a stream of image data.
- Tagged Image File with Wavelet CMP Compression.

BMP formats

- Windows .BMP, with no compression.
- Windows .BMP, with RLE compression.
- OS/2 .BMP Version 1.*x*.
- OS/2 .BMP Version 2.*x*.
- Wireless bitmap file. Type 0.

WMF and EMF formats

- Windows MetaFile.
- Windows Enhanced MetaFile.

Exif formats

- Exif file containing a TIFF image, with no compression and with RGB color space.
- Exif file containing a TIFF image, with no compression and with YCbCr color space.
- Exif file containing a JPEG compressed image.

1-Bit FAX formats

- TIFF, compressed by using CCITT.
- TIFF, compressed by using CCITT, group 3, 1 dimension.
- TIFF, compressed by using CCITT, group 3, 2 dimensions.
- TIFF, compressed by using CCITT, group 4.
- Raw FAX, compressed by using CCITT group 3, 1 dimension.
- Raw FAX, compressed by using CCITT group 3, 2 dimensions.
- Raw FAX, compressed by using CCITT group 4.
- IOCA, compressed using by CCITT group 3, 1 dimension.
- IOCA, compressed using by CCITT group 3, 2 dimensions.
- IOCA, compressed using by CCITT group 4.
- IOCA, uncompressed, with the MO:DCA wrapper.
- IOCA, compressed by using IBM MMR, with the MO:DCA wrapper.
- IOCA, uncompressed, with the MO:DCA wrapper.

Other 1-Bit formats

- MacPaint.
- Portable bitmap. ASCII File.
- Portable bitmap. Binary File.
- XBitmap File.
- Microsoft Paint.

PDF formats (Image and .PDF Files license required)

- PDF 1.3.

NOTE In the License Manager, you must turn on the PDF Enhanced level of licensing in addition to the base-level license if you want the AutoStore software to support .PDF file formats.

Other formats

- PS files (Image and .PDF Files license is required).
- Encapsulated PostScript (EPS) (Image and .PDF Files license is required).

Knowledge Package Loader component

Use the Knowledge Package Loader component to unpack .XML files that the AutoStore schema specifies.

Feature Highlights

The Knowledge Package Loader component features allow you to perform the following tasks.

- Determine whether or not to send the field values of the .XML document to the next component in the process.
- Configure any remaining components within your process by selecting each component and clicking on the **Configure** button.

The Knowledge Package Loader component is a blocking component. To add components to the process, select the component that you want to add, and then click **Configure** to configure that component.

This blocking component is typically used with Knowledge Package Builder. The builder component builds the .XML files, and then this component uses the .XML files.

Using the Knowledge Package Loader component

The following are examples of how to use the Knowledge Package Loader component.

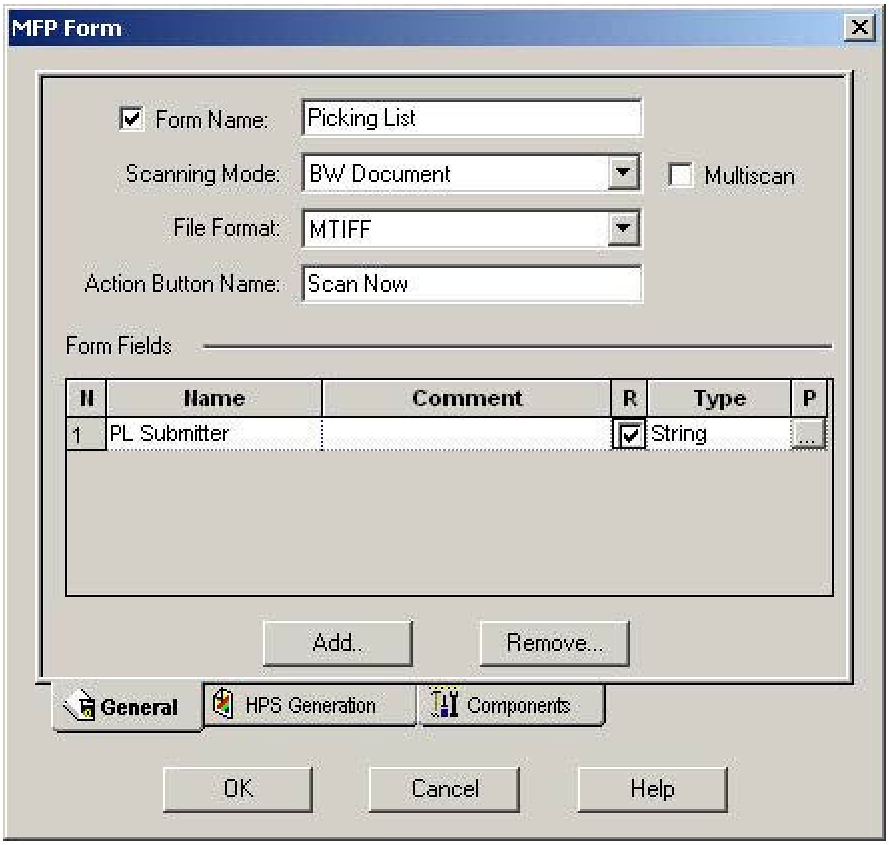
Case 1: An .XML file containing the image files and metadata can be placed in a directory (for example, by another AutoStore process using the Knowledge Package Builder component to build an .XML file). A second AutoStore process can then use the Poll Directory component to read the .XML files and use the Knowledge Package Loader process component to extract image files from the .XML files and process and route these image files using the XML field data included in the .XML file. This field data can be referenced by using the RRTs included in this component. By using the Knowledge Package Builder and Knowledge Package Loader components in this way, you can wrap and unwrap the AutoStore files and metadata into packages that can be transferred between processes or servers.

Case 2: To send a file to a client computer, wrap the file by using the component, and then send the file to the AutoStore queue. The client computer detects that something is in the queue and can use the Knowledge Package Loader component to unwrap the file for further processing.

NOTE Do not place a Knowledge Package Loader Process component immediately after a Knowledge Package Loader Capture component unless you ensure that the Capture component is going to generate an .XML file to be processed by the Process component.

The following examples describe how to use the Knowledge Package Builder and Knowledge Package Loader components with an MFP to encapsulate information into the .XML data format and to extract information from the .XML data format. The benefit of using the .XML data format in AutoStore processes is that the format can contain image data and also store user-defined information that can be retrieved when the data is scanned.

1. Load an AutoStore configuration file into an MFP that contains form fields and schema data that can be modified by using the MFP.



☑ Form Name: Picking List

Scanning Mode: BW Document ☐ Multiscan

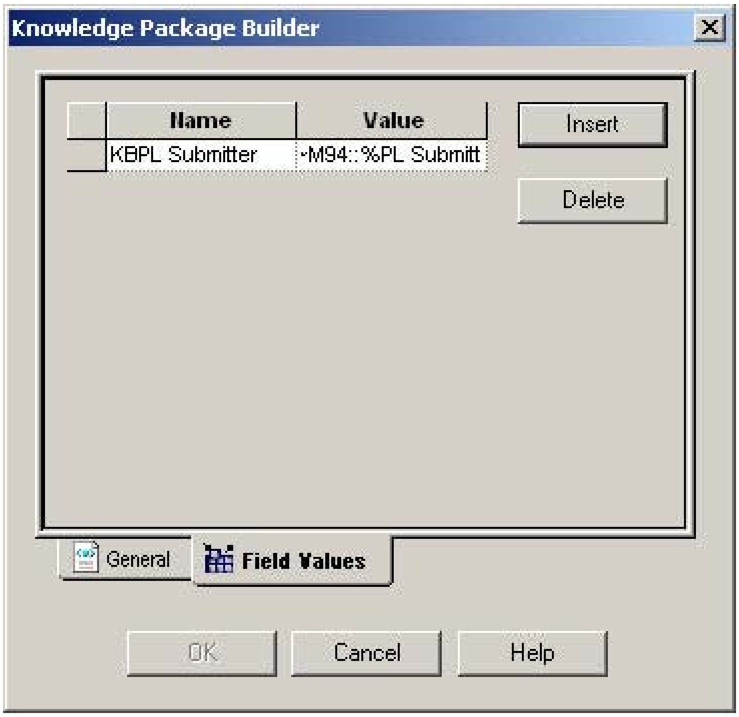
File Format: MTIFF

Action Button Name: Scan Now

Form Fields

N	Name	Comment	R	Type	P
1	PL Submitter		☑	String	...

The configuration file also contains a Knowledge Package Builder component that is used to generate the .XML data. The administrator has added form fields within the Knowledge Package Builder component that capture the data from the MFP forms.



Name	Value
KBPL Submitter	~M94::%PL Submitt

2. At the MFP, use the AutoStore **Send To** menu to enter data for the form fields that were generated by the AutoStore configuration file. When you are finished, press **Scan Now**, which is the **Action Button Name** that was designated in the configuration file.

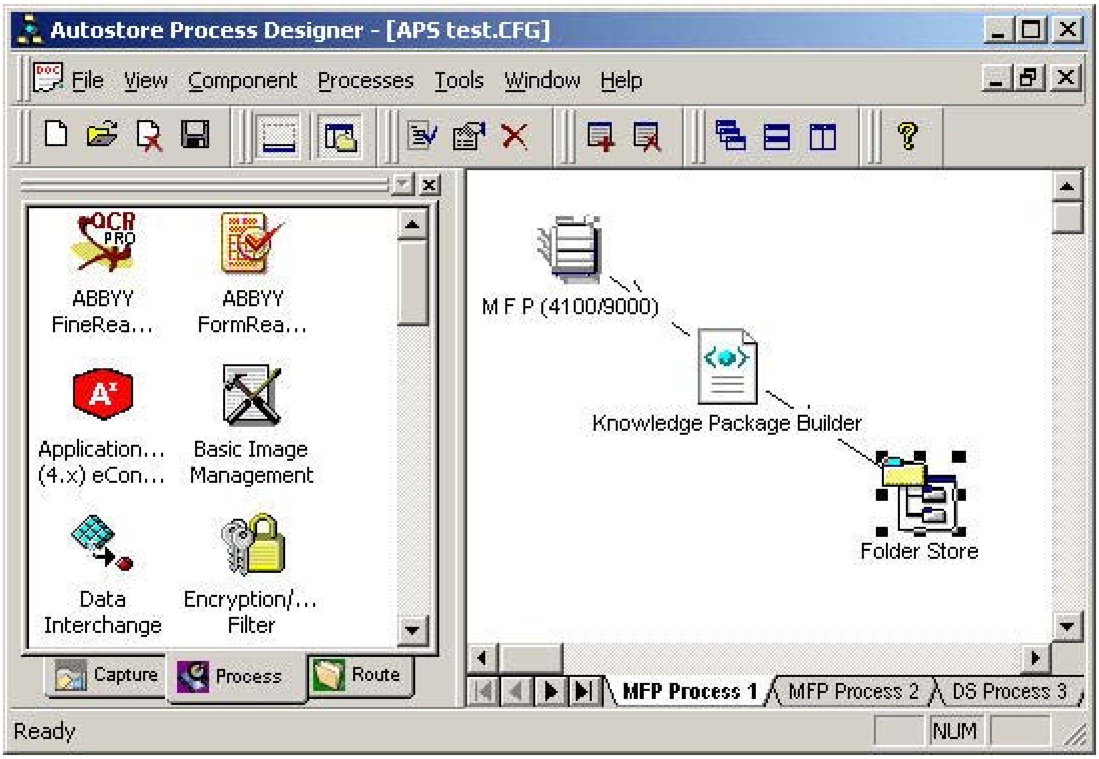
3. The MFP collects the MFP user-specified information as .XML metadata, scans the image, and then sends all of the information to the AutoStore server.

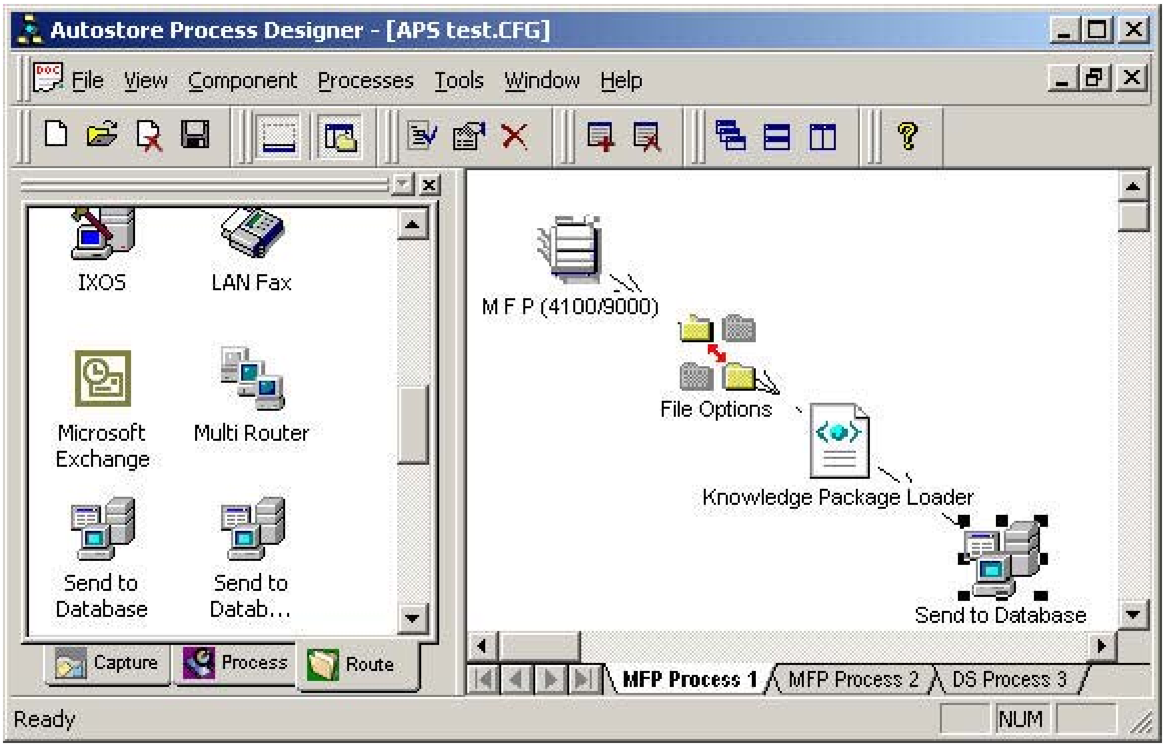
4. The AutoStore server starts the Knowledge Package Builder component, combines the metadata information and the image information into a single .XML file, and then pushes this information through the rest of the AutoStore process.

In a chain process, or on a different AutoStore server, the AutoStore administrator sets up additional process configuration tabs that contain a Poll Directory component that collects the .XML file and feeds the file into the Knowledge Package Loader component.

The Knowledge Package Loader component separates the metadata from the image. At this point, the metadata that was captured at the MFP as user-specified information is made available, along with the image, to the new AutoStore process.

The following two images are examples of the process and chain process that use the Knowledge Package Builder and Knowledge Package Loader components.





The metadata information is collected from the MFP in the form of the Submitter Name (PL Submitter). Through the AutoStore process, the metadata information is processed in .XML format and sent to the Chain process so that the Knowledge Package Loader component can extract and decode the .XML data and make the metadata available to the rest of the chain process. The metadata can be stored in a database.

The Knowledge Package Builder and Knowledge Package Loader components allow you to collect user-defined metadata from an MFP at the time of an image scan. Then, you can use this metadata in other chain processes, within the AutoStore server, or other knowledge object software applications. Without these two components, the metadata is lost after the first process completes.

Configuring the Knowledge Package Loader component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Knowledge Package Loader component.

The following attributes are available in the **Knowledge Package Loader** dialog box:

- **Component Name.** Lists the names of the components that are currently available in your process. You can click to select a component, and then click the **Configure** button to configure that component.
- **Description.** Provides a brief description of the component name.
- **Configure.** Click the **Configure** button to configure the selected component.
- **Include Fields.** Select this check box to include the field values from the .XML file into the knowledge object. The knowledge object is the memory record that contains the file and metadata for an AutoStore job. The knowledge object for a specific AutoStore job is created when the job is captured by the Capture component and contains all of the metadata associated with the AutoStore job. Knowledge Objects are envelopes that AutoStore creates for capturing content payload. Within each envelope RRTs are used to create tagged information for replacement by designated components. The components exchange data with each other through replacement of tagged variables with actual metadata content within the knowledge object envelope. Note that all of the fields that are loaded into the knowledge object are automatically saved into the Route component record when the destination data storage has a matching field name. As a result of checking this parameter, you are directly saving all of the .XML file field values into your Route component record.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **Knowledge Package Loader** dialog box and configure the Knowledge Package Loader component.

Using Digital Sender to configure the Knowledge Package Loader component

1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).

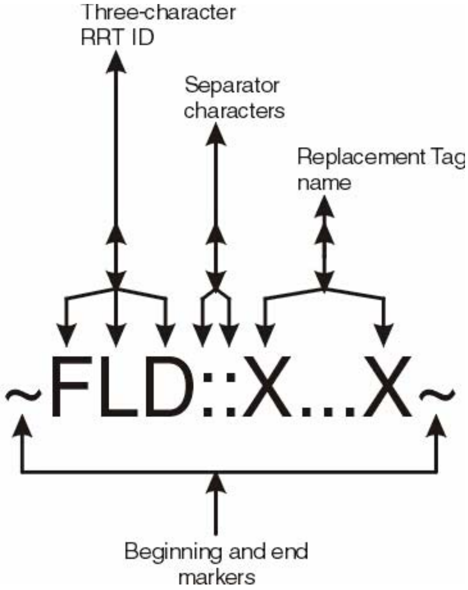
4. In the **Component Name** window, select the Knowledge Package Loader component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.

Segment name	Description
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **ASX**.

The following table describes the Reserved Replacement Tag Names (RRTN) values for this component.

Reserved Replacement Tag Names (RRTN)

Name	Description
Version	The version number of the .XML schema.
DateCreated	The date that the .XML file was created.
TimeCreated	The time that the .XML file was created.
AUTHOR	The field where you type the author's name.
COMMENTS	The field where you type comments into the file.

Field Replacement Tag Names

This component does support Field Replacement Tag Names (FRTNs) for field names contained within the .XML file. The following is an example on an FRTN.

~ASX::%Client ID%~ is replaced by "Hewlett-Packard" if the user types in `Hewlett-Packard` for the field name "Client ID"

Special Set Replacement Tag Names (SSRTN)

No SSRTN is supported by this component.

Troubleshooting tips

Problem Description	Solution
You cannot configure a component because it does not appear on the list of components on the **Component** tab.	You must add the component to the AutoStore process or it does not show up in the Knowledge Package Loader blocking component.

(continued)

Problem Description	Solution
A "Duplicate Output destination" error appears when you attempt to run your AutoStore process.	If the **Include Fields** check box is selected and the next component in the process (for example, Send to Database) is using RRT ASX, the process will fail with a database error.

Restrictions and limitations

Do not place a Knowledge Package Loader Process component immediately after a Knowledge Package Loader Capture component unless you ensure that the Capture component is going to generate an .XML file to be processed by the Process component.

Knowledge Package Builder component

This component encapsulates all of the information that is related to a batch job into an .XML file. The XML schema provides the most effective means of transferring the complete batch job information between processing centers, such as from a remote site to central servers, between a workstation and a server, or between two .XML-enabled programs.

This component creates an .XML file anywhere within the capture process. When an .XML file is generated, it can be stored or transferred to the receiving program using any of the available Route components.

The Knowledge Package Builder component uses the XML schema that is published within the published, open SDK documentation.

Feature highlights

This component features the dynamic creation of .XML files, a document and image file encapsulation option, an unlimited index field, and dynamic and static field-value support.

- Select the check box to activate this component within a selected process.
- Supply the document name, the author, and any comments that you want to add to the document.
- You can embed the document. For example, if the documents are .TIF files, you can embed the .TIF files into the .XML file.
- Select the **Include Field Values** check box to include the field values from the knowledge object.
- If you do not want to send the original document as a separate file outside of the .XML file (for example, the .TIF file), you can choose to not send it.

The most common input file types for this component are .PDF files and Microsoft Word documents. This component is widely used with the PDF Converter component, and any Route component. Use this component in conjunction with BizTalk server to send documents and data into a BizTalk process.

Using the Knowledge Package Builder component

Use the Knowledge Package Builder component to encapsulate all of the content within a process into an .XML document and communicate with other XML-enabled processes.

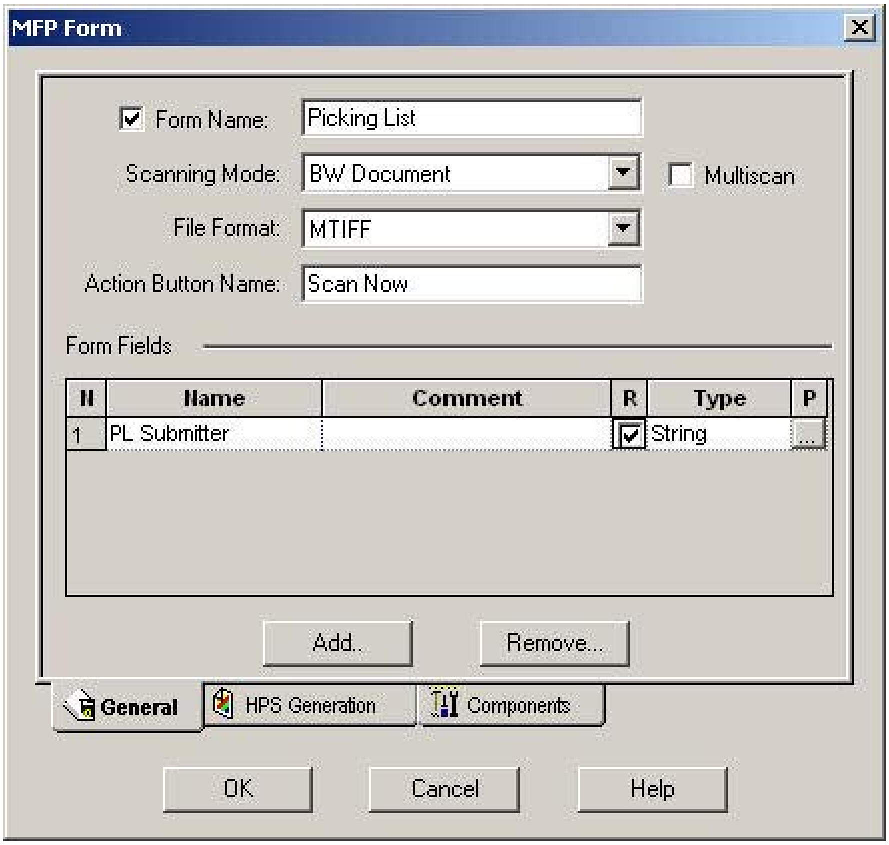
You can also use this component to communicate between two Knowledge-Package-enabled products (HP DSS Workflow and AutoStore). The Knowledge Package Builder component packs all of the information into an .XML document, and then the Knowledge Package Loader component takes a packed .XML document and loads it into the process stream. Use these two components to build, send, and load complete job batches between servers and processes. The following items are examples of how you can use this component:

- **Interprocess communication.** Use the Knowledge Package Builder component to generate .XML files and have central servers read and process the queue Knowledge Objects.
- **Load Balancing.** Use the Knowledge Package Builder component to send jobs between two servers and distribute the load.
- **Workstation communication.** Generate .XML files (by using Folder Store or Content Management Queues) so that the other Knowledge Package Loader component-enabled programs can read and process the queue.
- **Distributed Process Chaining.** Create chained processes (within one server or multiple servers) that run as one integrated process by exchanging XML messages.
- **External Application Messaging.** Use XML messaging to connect your processing server and station to external XML messaging platforms such as MS BizTalk. Use MS BizTalk with your server to gain access to all types of back-end programs.
- **Complete Batch Job Encapsulation.** Each XML message is designed to include a full and complete set of batch-job information, and contains options including user-defined fields, index data, and all attached files.
- **Distributed Load Processing.** Use XML to distribute the load to multiple servers and process batch jobs in a distributed environment.

This published XML interface schema is designed for program integration. The XML schema allows for direct and clear encapsulation of files and distribution of information. See the *Software Development Kit for XML Schema Definitions*, which is available through the NSi Software Developer's Partnership Program.

The following examples describe how to use the Knowledge Package Builder and Knowledge Package Loader components with an MFP to encapsulate information into the .XML data format and to extract information from the .XML data format. The benefit of using the .XML data format in AutoStore processes is that the format can contain image data and also store user-defined information that can be retrieved when the data is scanned.

1. Load an AutoStore configuration file into an MFP that contains form fields and schema data that can be modified by using the MFP.



MFP Form

☑ Form Name: Picking List

Scanning Mode: BW Document □ Multiscan

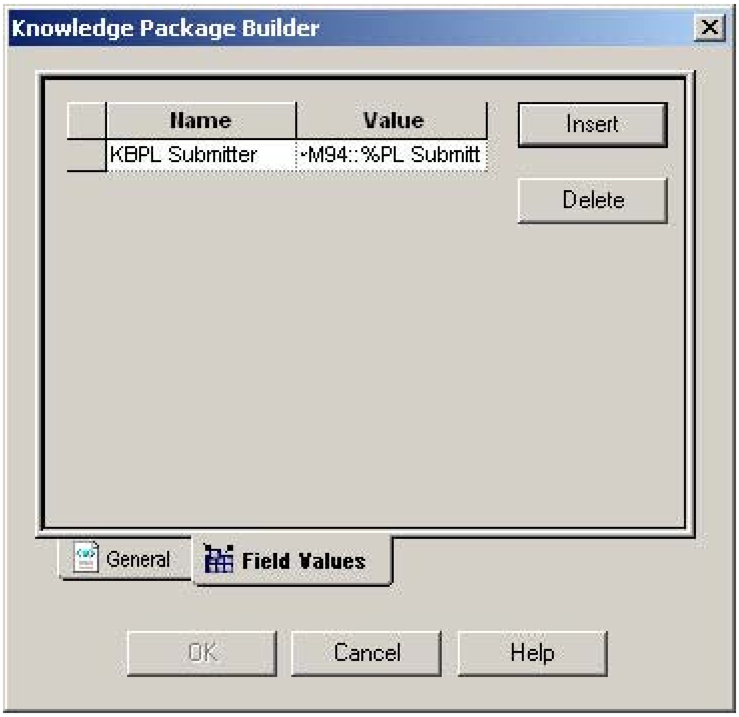
File Format: MTIFF

Action Button Name: Scan Now

Form Fields

N	Name	Comment	R	Type	P
1	PL Submitter		☑	String	...

The configuration file also contains a Knowledge Package Builder component that is used to generate the .XML data. The administrator has added form fields within the Knowledge Package Builder component that capture the data from the MFP forms.



Name	Value
KBPL Submitter	~M94::%PL Submitt

2. At the MFP, use the AutoStore **Send To** menu to enter data for the form fields that were generated by the AutoStore configuration file. When you are finished, press **Scan Now**, which is the **Action Button Name** that was designated in the configuration file.

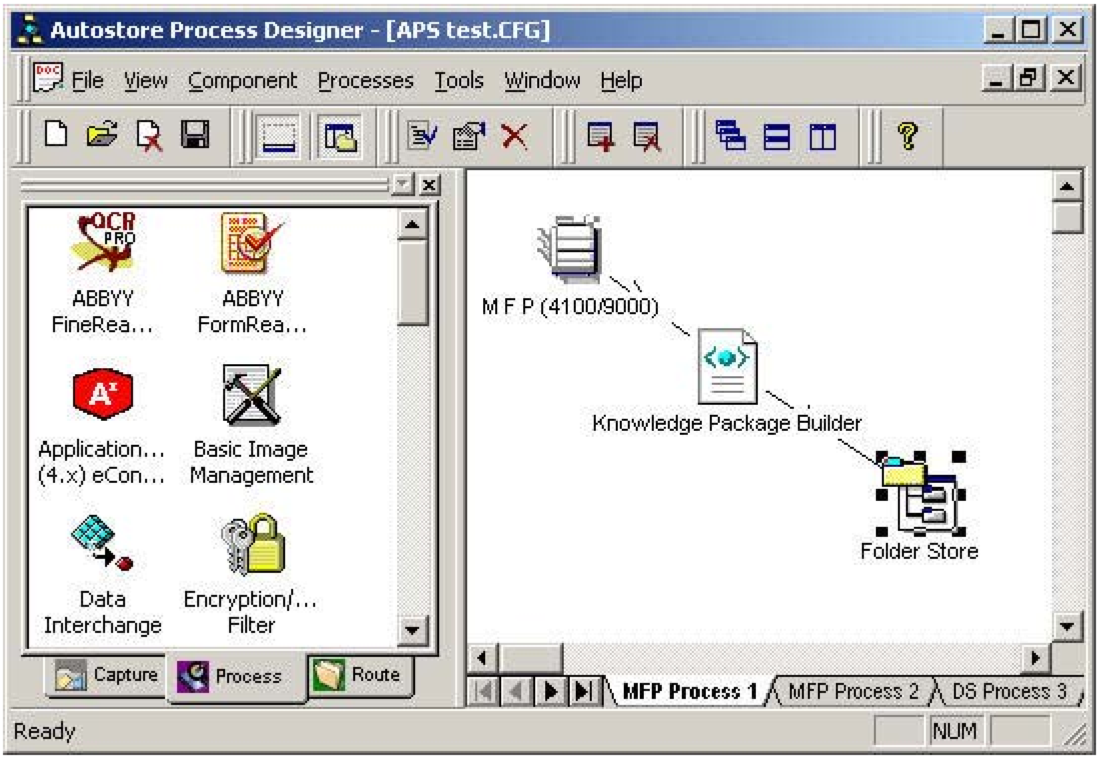
3. The MFP collects the MFP user-specified information as .XML metadata, scans the image, and then sends all of the information to the AutoStore server.

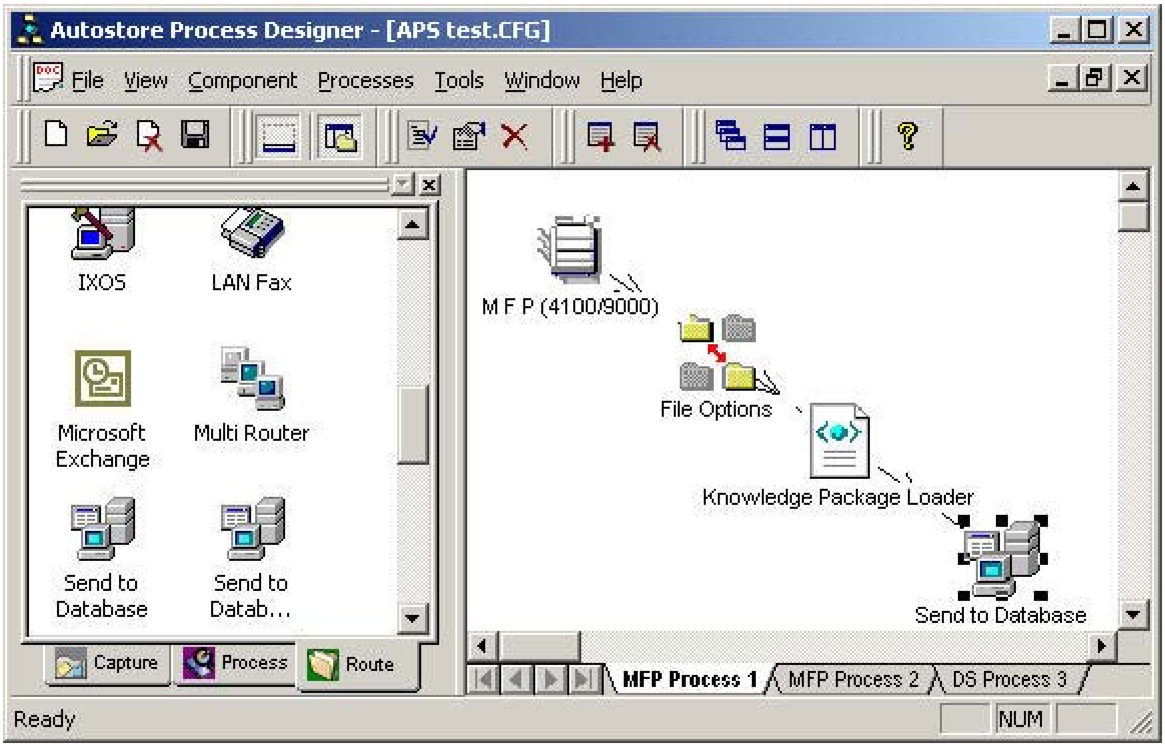
4. The AutoStore server starts the Knowledge Package Builder component, combines the metadata information and the image information into a single .XML file, and then pushes this information through the rest of the AutoStore process.

In a chain process, or on a different AutoStore server, the AutoStore administrator sets up additional process configuration tabs that contain a Poll Directory component that collects the .XML file and feeds the file into the Knowledge Package Loader component.

The Knowledge Package Loader component separates the metadata from the image. At this point, the metadata that was captured at the MFP as user-specified information is made available, along with the image, to the new AutoStore process.

The following two images are examples of the process and chain process that use the Knowledge Package Builder and Knowledge Package Loader components.





The metadata information is collected from the MFP in the form of the Submitter Name (PL Submitter). Through the AutoStore process, the metadata information is processed in .XML format and sent to the Chain process so that the Knowledge Package Loader component can extract and decode the .XML data and make the metadata available to the rest of the chain process. The metadata can be stored in a database.

The Knowledge Package Builder and Knowledge Package Loader components allow you to collect user-defined metadata from an MFP at the time of an image scan. Then, you can use this metadata in other chain processes, within the AutoStore server, or other knowledge object software applications. Without these two components, the metadata is lost after the first process completes.

Licensing for the Knowledge Package Builder component

Three types of licenses are available for this component:

- **Evaluation.** A 30-day fully functional component is available upon first installation.
- **Licensed.** The fully licensed component provides full capabilities indefinitely.
- **Expired.** After the evaluation period, unlicensed components expire without any further processing.

Configuring the Knowledge Package Builder component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Knowledge Package Builder component.

Depending on which Capture component you are using, follow the appropriate procedure to open the **Knowledge Package Builder** dialog box and configure the Knowledge Package Builder component.

The following attributes are available in the Knowledge Package Builder dialog box:

General tab

Specify the general .XML file attributes on the **General** tab.

- **Activate.** Select this check box to activate this component on each process within blocking components. Multifunctional devices or similar Capture components use this feature to activate this component based on your input.
- **File name.** Type the name of the .XML file that is to be generated. If you leave this attribute field blank, a system-generated file is used.

- **Author.** Type the name of the author of the XML document.
  - **Comments.** Type a description of the document or any other applicable comment.
  - **Embed documents.** Select this check box to embed the files into the XML document. All documents are converted to ASCII characters. The correct conversion from ASCII to binary or other format must be performed by the XML document recipient.
  - **Include field values.** Select this check box to include field values from the knowledge object in the XML schema. **Include field values** includes the knowledge object fields in the .XML file. However, if you want to include additional fields in the .XML file, you can do so by clicking the **Field Values** tab and typing a name and value. Use field values to code or index your XML documents based on your process parameters.
- **Delete original documents.** Select this check box to remove the original documents after the XML document has been created. If you do not delete original documents, then your process will add the XML document as a new file to the list of existing files.

Field Values tab

You can add fields to or delete fields from the .XML file by configuring attributes on the **Field Values** tab.

- **Insert.** Use this button to add or delete fields to the XML document by configuring attributes on this tab.
- **Delete.** Use this button to remove a field value pair from the list of field value entries. The field will no longer be a part of the XML document.

Using Knowledge Package Loader to configure the Knowledge Package Builder component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Knowledge Package Builder component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Knowledge Package Builder component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Knowledge Package Builder component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Knowledge Package Builder component

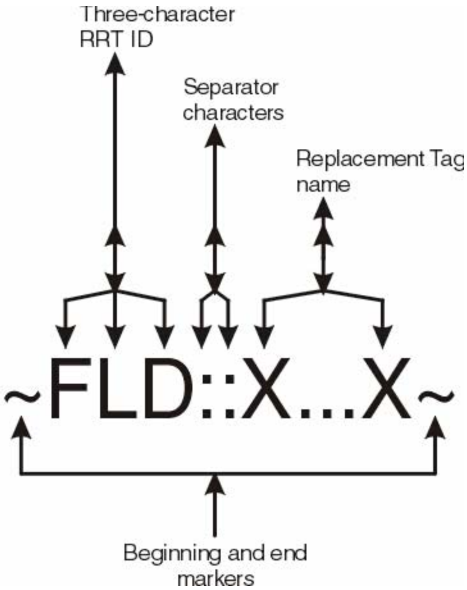
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Knowledge Package Builder component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs, however this component's parameters can contain RRT strings.

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

- When the field name contains only one single character and the **Embed Documents** option is *not* selected, the .XML file is not generated. However, the Status Monitor indicates that the .XML file was created successfully. Do not use a field name that contains only one character.

ABBYY FineReader OCR v6.0 component

The ABBYY FineReader OCR v6.0 component uses an omnifont optical-text-recognition system. This system recognizes practically any font, without prior training. One of this component's features is its high recognition accuracy and low sensitivity to print defects. These features are the result of a special recognition technology that is based on the principles of Integral Purposeful Adaptive (IPA) perception, which is fully implemented in Professional OCR.

Use the ABBYY FineReader OCR v6.0 component when your process requires font-independent adaptive OCR with a high accuracy rate. This component can also produce searchable .PDF files by using the .PDF standard-file definitions.

Feature highlights

ABBYY FineReader OCR v6.0 is a Process component that can accept image files as input. The primary use of this component is to convert searchable data files into one of seven supported formats. This component also provides a more accurate OCR quality when font variation exists among imaged documents.

You can use various formatting and detection parameters to optimize ABBYY FineReader OCR v6.0 for your specific needs.

The ABBYY FineReader OCR v6.0 component offers the following features.

- High recognition accuracy and low sensitivity to print defects. These features are the result of a special recognition technology that is based on the principles of IPA perception, which is fully implemented in Professional OCR.
- No page counting. The ABBYY FineReader OCR v6.0 component does not require hardware keys or any type of page counting for .PDF file conversion.
- Accepts image files as input.

Using the ABBYY FineReader OCR v6.0 component

Set the component attributes that your business process requires and add a Route component to the end of your process. The image files are automatically converted to the required output file format and then are handed to the route (destination) component.

Licensing for the ABBYY FineReader OCR v6.0 component

Converting ABBYY FineReader OCR v6.0 output as a .PDF file requires a valid license.

Configuring the ABBYY FineReader OCR v6.0 component

Use static or dynamic values as defined in the Capture component runtime replacement tags (RRTs) to set the case-sensitive attributes for the ABBYY FineReader OCR v6.0 component.

The following attributes are available in the **ABBYY FineReader OCR** dialog box.

General tab

Use the **General** tab to select the general ABBYY FineReader OCR v6.0 attributes.

- **Activate.** Select this check box to activate OCR processing. You can select this check box to activate the ABBY FineReader OCR v6.0 component based on the process setting within a blocking component.
- **Delete Original Image.** Select this check box to delete the image after processing.
- **Language.** Select from 65 available languages. If required, you can select multiple languages by using a comma to separate the language names.
- **Output OCR Text as**. Select from the following options:
  - **File.** Select from different file types (for example, .TXT, .XLS, or .PDF) as the preferred file type for saving the OCR text.
  - **Runtime Replacement.** Use the replacement tag, **~FRO::OCRText~**, to save the OCR text in a field.
- **Document Type.** Select from the following options:
  - **Auto detect layout.** Indicates that text layout is detected automatically. This value is set by default. If this value is set, all text types, including multi-column texts, texts with tables, and pictures, are recognized automatically.
  - **Single column.** Indicates that the text is formatted in one column. Use this option if auto detect layout incorrectly determined a multi-column text type.
  - **Plain text formatted with spaces.** Indicates that the text is formatted in one column and set in a monospace font of a same size. In the recognized text, left indents are represented as spaces. Every line is made into a separate paragraph and empty lines separate the original paragraphs. Use this type if you expect C++ code printouts or old computer printouts.
- **Print type.** Select from the following options:
  - **Auto detect.** Automatically detect the print type.
  - **Typewriter.** Select this when documents are typewriter-printed documents to improve the recognition rate.
  - **Dot matrix printer.** Select this when all of your documents are dot-matrix-printer printed documents.

- **Tables.** In most cases, the program automatically divides the table into rows and columns. You can fine-tune tables by using the following options:
  - **One line of text per cell.** Use this option if the table lacks black separators, or some separators are black and others are not. The table should only have one line of text per cell.
  - **No merged cells in table.** Use this option if the table has no merged cells in it.
- **Image Processing.** Select one or both of the available image processing options.
  - **Detect Image Orientation.** Select this check box to detect the page orientation during layout analysis. If the page orientation differs from normal, the image is automatically rotated.
  - **Detect Inverted Image.** Select this check box to detect whether the image is inverted (white text against a black background). The text color is detected during layout analysis. If the text color differs from normal, it is automatically inverted (changed to black text against a white background).

NOTE Do not select **One line of text per cell** and **No merged cells** in table options if your text contains tables of different structures. Selecting these options can cause layout analysis mistakes and negatively affect the recognition quality.

Formatting tab

Set up the format of the processed file by using the options that are defined in the **Formatting** tab.

- **Retain layout.** Select from the following options:
  - **Retain full page layout.** This option retains the layout in full: arrangement into paragraphs, font and font size, columns, text direction, text color, and the structure of tables.
  - **Retain font and font size.** This option retains the structure of tables, arrangement into paragraphs, font, and font size.
  - **Remove all formatting.** This option retains only the structure of tables and arrangement into paragraphs.

NOTE Some advanced options depend on the export format you choose. For example, with .RTF and .DOC formats you can set the default page size and uncertain character highlight mode; with .HTML file format, you can set the picture resolution and the code page; with .PDF file format, you can set which Type 1 fonts to use (if you use a code page different from the Latin – for example, Cyrillic, Greek, and so on). The dialog box has a separate tab for each format. Just click the format tab that you need and set the options that you want.

- **Keep pictures.** This option saves the picture together with the recognized text. This option is available if you save in .RTF, .DOC, or .HTML file format.
- **Fonts.** Choose the serif, sans serif, or mono-spaced fonts.
- **Format Settings.** Click the **Format Settings** button to further select the file type format of the OCR text. From the available file types, you can choose the file type of the OCR text. When you click any of the seven available tabs, you can select the parameters for that file type. By default, the fonts from the **Formatting** tab are used when you save in .RTF, .DOC, or .HTML file format.

RTF/Doc tab

Field name	Description
Default paper size	Sets the paper size for saving in the .RTF or .DOC file format. If your pages do not fit in this size, the program will set the paper size automatically.
Remove optional hyphens	Removes the optional hyphen sign (¬) from the recognized text. If the **Keep line breaks** option is set, hyphen signs (-) replace the optional hyphen signs.
Keep line breaks	Set this option if you want the original arrangement of lines to be retained in the .RTF or .DOC file format.
Retain text color	Set this option if you want the original character color to be retained. **NOTE** Microsoft Word Versions 6.0, 7.0, and 97 (Version 8.0) have a limited text and background color palette. The colors you choose might be replaced with colors from the Word palette. Word 2000 (Version 9.0) uses the color that you set.
Keep page break	Set this option if you want the original document page arrangement to be retained in either the .RTF or .DOC format.
Highlight uncertain characters with text color	Sets the character color that is used to highlight uncertain characters. **NOTE** Microsoft Word Versions 6.0, 7.0, and 97 (Version 8.0) have a limited text and background color palette. The colors you choose might be replaced with colors from the Word palette. Word 2000 (Version 9.0) uses the color that you set.
Highlight uncertain characters with background color	Sets the background color that is used to highlight uncertain characters. **NOTE** Microsoft Word Versions 6.0, 7.0, and 97 (Version 8.0) have a limited text and background color palette. The colors you choose might be replaced with colors from the Word palette. Word 2000 (Version 9.0) uses the color that you set.

PDF tab

Field name	Description
Save mode	Select one of the following options: ● Text and pictures only The recognized text is saved as text, and the pictures are saved as pictures. The original document design (font, background, and layout marking) is not retained. ● Text over the page image The entire image is saved as a picture. Text areas are saved as text over the picture. The entire image is saved as a picture. Text areas are saved as text over the picture. ● Text under the page image The entire image is saved as a picture. The recognized text is placed under it. This option is useful if you export your text to document archives: the full-page layout is retained, and the full-text search is available if you save in this mode.
Retain text and background color	Set this option if you want the original background and character color to be retained. If the "Text under the page image" mode is on, text and background color are saved automatically.
Replace uncertain words with image	If you save your document in the "Text and pictures only" or the "Text over the page image" mode, words that cannot be recognized correctly might be replaced with their images. Set this option.

PDF tab (continued)

Field name	Description
Font use mode	Use Acrobat Reader standard fonts. ● The .PDF file refers to the standard system fonts: Times, Helvetica, and CourierNew. Refer to Type 1 fonts. ● Only references to the Type 1 fonts are registered in the .PDF file, the fonts themselves are not embedded. Fonts that the .PDF file references must be installed and available at the Adobe Type Manager. Embed Type 1 fonts. ● The Type 1 fonts are embedded in the .PDF file. **NOTE** In the Refer to Type 1 fonts mode, only references to the fonts in use are registered in the .PDF file (the fonts themselves are not actually embedded in the .PDF file as they are in the Embed Type 1 fonts mode). This means that the .PDF file occupies less disk space than the same files saved in the Embed Type 1 fonts mode. Saving text in the "Embed Type 1 fonts" mode allows other users to view, edit, and print the document in the original fonts even if these fonts are not installed on their computers.

HTML tab

Field name	Description
Code page	By default, the code page is detected automatically. Select the Auto value to use the automatic detection. You can select the code page manually, if necessary: simply select the value that you need from the list.
Code page type	Specify the code page (Windows, MS-DOS, Macintosh, ISO).
Picture resolution	Specify the picture resolution to be used in the format. In most cases, the default resolution of 72 dpi is sufficient. **NOTE** Pictures are saved as separate .JPG files.

HTML tab (continued)

Field name	Description
Keep line breaks	Set this option if you want the original arrangement of lines to be retained as the format. Otherwise, the text is formatted in a single line in the .HTML file.
Keep text color	Set this option if you want the original character color to be retained.
Use solid line as page breaks	The original arrangement of pages is retained, and pages are separated by a solid line.
Format (uses CSS, requires Internet Explorer 4.0 or later)	● Full In this option, the new .HTML file format (HTML 4) is used. It supports any type of document-layout retention. (The actual retention type used depends on the options that are set on the **Formatting** tab in the Retain layout group.) The built-in style sheet is used. ● Simple In this option, the .HTML 3 file format is used. The document layout is retained approximately: first-line indent and indents in tables are not retained. All browsers (including Netscape Navigator, Internet Explorer 3.0 and later support this format). ● Auto In this option, both formats (Simple and Full) are saved to the same file. The browser that you use selects the format that it supports.

TXT tab

Field name	Description
Code page	By default, the code page is detected automatically. Select the **Auto** value to use the automatic detection. You can select the code page manually, if necessary. Simply select the value that you need from the list.
Code page type	Specifies the code page (Windows, MS-DOS, Macintosh, ISO).
Keep line breaks	Set this option if you want the original arrangement of lines to be retained in the .TXT file format; otherwise, the text will be formatted in a single line in the .TXT file.
Append to the end of file	Appends the text to the end of an existing .TXT file.

TXT tab (continued)

Field name	Description
Insert page break character (#12) as page break	Set this option if you want the original document page arrangement to be retained in .TXT file format.
Use blank line as paragraph separator	Set this option if you want the paragraphs to be separated by blank lines in the .TXT file.

CSV tab

Field name	Description
Code page	By default, the code page is detected automatically. Select the **Auto** value to use the automatic detection. You can select the code page manually, if necessary: simply select the value that you need from the list.
Code page type	Specifies the code page (Windows, MS-DOS, Macintosh, ISO).
Field separator	Specifies the character that separates the fields in the .CSV file.
Ignore text outside tables	Set this option if you only want to save tables in the .XLS file.
Append to the end of file	Appends the text to the end of an existing .CSV file.
Insert page break character (#12) as page break	Set this option if you want the original document page arrangement to be retained in .CSV format.

DBF tab

Field name	Description
Code page	By default, the code page is detected automatically. Select the **Auto** value to use the automatic detection. You can select the code page manually, if necessary: simply select the value that you need from the list.
Code page type	Specifies the code page (Windows, MS-DOS, Macintosh, ISO).
Append to the end of file	Appends the text to the end of an existing .DBF file.

XLS tab

Field name	Description
Ignore text outside tables	Set this option if you only want to save tables in the .XLS file format.
Convert numeric values to numbers	Set this option to save numbers in the Numbers format in the .XLS file. Microsoft Excel can perform arithmetical operations with cells of this format.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **ABBYY FineReader OCR** dialog box and configure the ABBYY FineReader OCR v6.0 component.

Using Knowledge Package Loader to configure the ABBYY FineReader OCR v6.0 component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the ABBYY FineReader OCR v6.0 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the ABBYY FineReader OCR v6.0 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the ABBYY FineReader OCR v6.0 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the ABBYY FineReader OCR v6.0 component

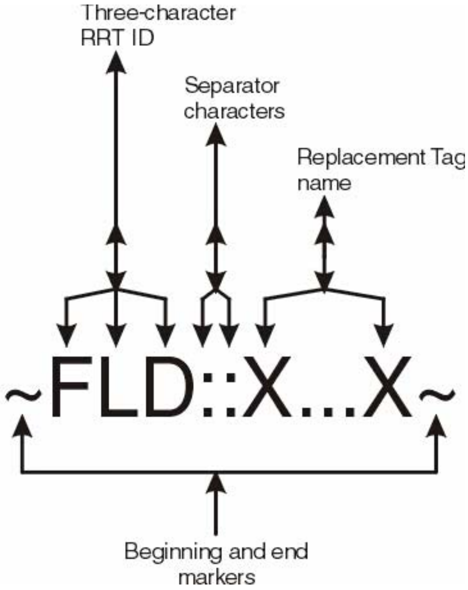
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the ABBYY FineReader OCR v6.0 component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and document time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

NOTE ABBYY FineReader OCR v6 does not generate RRTs; however, all of the attributes can contain RRTs.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FRO**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for this component.

Name	Description
OCRText	The text results from the Fine Reader OCR engine.

The following is an example of the RRTN process:

~FRO::OCRText~ - This RRTN is replaced with the OCR text of the document from the FineReader OCR engine.

Field Replacement Tag Names

This component does not generate Field Replacement Tag Names (FRTNs). However, all of the parameters can contain RRTs from other components. For example, when the HP LaserJet 4100mfp or HP LaserJet 9000mfp is the Capture component, the "LANGUAGE" parameter can be set to "~M94::%Language%~". Using this convention, the function keys that have the field name Language can drive the OCR language selection.

Special Set Replacement Tag Names (SSRTN)

This component does not generate SSRTNs. However, all of the parameters can contain RRTs from other components.

Troubleshooting tips

Problem	Solution
An error occurs in processing the images when using OCR. The error code in the Status Monitor appears as "Error OCR'ing this image. Error code: 8xxx".	The input file format is not a supported file format. Review the files in your process and make the appropriate changes.

Problem	Solution
You are seeing poor-quality OCR results.	Inaccuracies in the OCR process can have many causes. It is recommended that you perform an analysis of types of paper, scanners, and resolution levels to optimize your OCR results before setting up OCR processes. The following are some common tips for increasing OCR accuracy. 1. File format: Color documents do not capture image details accurately. When the process input is a color image, you achieve lower-quality OCR documents. Review your color document requirements and consider higher-resolution scanning to increase accuracy. 2. Document quality: Low-quality paper documents are another major cause of lower OCR accuracy. Lower-quality documents generally increase the error rate for OCR. When working with such documents, consider the following factors to increase your OCR accuracy: – Try to discover ways to get higher-quality paper documents. – Consider using a scanner that has a different scanner bulb color, which might work better with the paper color of your document. – Test a higher-level scan resolution. – Consider using the image process in advance of OCR to clean up the image.
Professional OCR randomly fails to produce good OCR-color uncompressed .TIF file images.	This problem was fixed in the 24 May 2002 version of the OCR filter integration (IFROCR.DLL). Symptoms can include accumulation of .TIF file images or .TMP files in the temp area under WINNT\TEMP or C:\TEMP or in the local settings area of an account profile. The system was not stable under the older design, especially when the service was running a multitask script that included more than one OCR task. Stop the service and update the server by installing the latest version of the service pack.

Problem	Solution
The OCR produces multiple language paragraphs in the same document.	This feature is only available through AutoStore 2002. With the Professional OCR module, you can only select one language (by default) through the AutoStore Process Designer (APD). In AutoStore 2002, you can activate this feature. You can change the language setting to be English, French, German; and so on. Use a comma as a separator in this setting.
The AutoStore server is performing slowly.	**NOTE** Both the image-processing filter and OCR engine use the physical memory and the processor intensively. The amount of memory and CPU speed that are required vary from one system to the other, depending not only on the workload but also on the average document size. If you are processing multiple small files (4 to 10 pages per document, at most), the following system configuration might be sufficient: One CPU PIII 600 with 512 M RAM or more running on a Windows 2000 server that has the latest service pack. If you have a larger workload and the documents are slowing down your server, you might need a more powerful configuration. Because the reasons behind this issue vary, the only way to isolate the bottleneck is to run one batch of images and monitor the following parameters on your Task Manager: 1. If CPU usage exceeds 90% for the duration of the processing, you need a more powerful processor. 2. If virtual memory exceeds 120% of the physical memory during the processing time, you need to add more physical memory. 3. If you have a large virtual memory file, your server will operate more slowly, even though the job will be completed.

Problem	Solution
The Renaming Schema does not work with either the HTML OCR or the WebStore.	The Renaming Schema does not work when you are using the OCR or the OCR PRO module along with SPS, Folder Store, or exchange store, or when using only the HTML WebStore. You cannot use the rename option because the image links within the generated .HTML file will be incorrect. The only way to use both is by manually adding the following line at the store module in each script: **EXCLUDE EXTENSIONS = GIF,JPG,JPEG** This command excludes the files with these extensions from being renamed, as shown in the following example: **OVER WRITE = NO** **Security = NO** **FOLDER PATH = c:\as\output\** **RENAME FILE = YES** **RENAME SCHEMA = Digital Document By ~Sender~ %c %e** **// adding the exclude extensions item** **EXCLUDE EXTENSIONS = GIF,JPG,JPEG** **// adding the exclude extensions item** **}** **POLL DIRECTORY = c:\as\input** **WORKING DIRECTORY = c:\as\work\** **HPS PASSTHROUGH = 0** **FILEEXT = HPS** **ON FAILURE = M** **ON SUCCESS = M**
The OCR does not convert images when the output is HTML.	When using schema with OCR to scan to an .HTML file, images appear as broken links. The schema renames the image when it is removed from the text body, before the OCR conversion. When the ABBYY FineReader OCR component is putting the .HTML file together, it does not look for the renamed schema image file (Document1). When no schema is used, the .HTML file is complete. When schema is used, the new file names are not expected.

Restrictions and limitations

- Supported file formats are .MTIFF 6.0 B/W (no color), .TIF images, B/W and Color (uncompressed) .BMP, and .JPG files.
- It is not possible to use the .PDF file format as input to the OCR engine.
- The recommended resolution is 200 to 300 dpi.
- The recommended file format for business documents is .MTIFF.
- The recommended color format is B/W.

SharePoint Portal v1.0 component

Use the SharePoint Portal v1.0 component to store documents into a centralized, unified interface for enterprise users and highly flexible deployment options.

The only difference between the SharePoint Portal v1.0 *Route* component and the SharePoint Portal v1.0 *Process* component is that the SharePoint Portal v1.0 *Process* component has two additional fields on the **Field Values** tab: **Field Passthrough** and **URL Field Name**. **Field Passthrough** and **URL Field Name** are not available for the Route component.

In an AutoStore process, the SharePoint Portal v1.0 component uses the SharePoint Portal Server (SPS) for document management. Use this component to store files in Microsoft SharePoint Portal Server v1.0.

Feature highlights

You can perform the following tasks by using the SharePoint Portal v1.0 component:

- Provide the general SharePoint Portal information (server, workspace, user name, password, folder path, content source, and workflow).
- Specify a location for storing files.
- Rename files that have duplicate names by using a schema name.
- Check-in files to allow other users to open and update them.
- Assign and change document attributes such as author, title, keywords, description, and categories.
- Repeat file names. The SharePoint Portal v1.0 component appends duplicate filenames with a counter. For example, if the original file name was TEST.TXT, the component rename schema can rename the files to TEST1.TXT, TEST2.TXT, TEST3.TXT, and so on.

Using the SharePoint Portal Process component

In an AutoStore process, the SharePoint Portal v1.0 component is frequently used with the Digital Sender component and the Poll Directory component. In this type of process, the SharePoint Portal v1.0 component picks up files and stores them in the SPS.

For example, if a file is stored in a folder that you want to share with the rest of the company, you can create a process that uses Poll Directory as the Capture component. Save the file that you want to share in a designated Input folder, and then use SharePoint Portal v1.0 as the Route component.

You can use the SharePoint Portal v1.0 component as a Process component only when the Route component in the AutoStore process can accept field values (such as Lotus Notes, open database connectivity [ODBC], and so on).

NOTE SharePoint Portal Server 2001 Client components, SPSCLIENT.MSI, must be installed on the AutoStore Server.

Configuring the SharePoint Portal v1.0 component

Depending on which Capture component you are using, follow the appropriate procedure to open **SharePoint Portal v1.0 Server** dialog box and configure the SharePoint Portal v1.0 component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the SharePoint Portal v1.0 component.

The following attributes are available in the **SharePoint Portal v1.0 Server** dialog box.

General tab

Use the options on this tab to specify where you want to save incoming files.

- **Server.** When you add the SharePoint Portal v1.0 component, you must specify which server will store the documents that this component processes.
- **Workspace.** The SharePoint Portal v1.0 component uses a workspace concept to provide access to document libraries, content sources, and categories. Select the workspace in which you want to store your information.
- **User Name.** Type a valid user name to log into the SharePoint Portal Server.
- **Password.** Type the password that corresponds to the user name to log into the SharePoint Portal Server.
- **Folder Path.** Select a folder path. Folders can have multiple folders within them. The **OK** button is available when you select a folder. A folder might already have other folders in it.

  You can create a folder dynamically by typing a new name for the folder.
- **Content Source.** You can add new content sources to the workspace so that you can gain access to content outside of the document library. Click on "**...**" to view a list of content source from which to select.
- **Workflow.** Select the document library where you want to work. Document libraries can have document libraries within them. Therefore, when you click the browse button (**...**) for **Document Library**, a tree view shows the document libraries hierarchy. You can create infinite document libraries inside document libraries if your computer has enough memory to support them.

  To enable the **OK** button, select a workspace. You can store documents only in the workspace, which is a subordinate document library. After you select the document library, you must then select the Folder Path for the document library.

Document Setting tab

Use the options on this tab to set the document attributes.

- **Author.** Type a valid author name.
- **Title.** Type the title of the document that you are going to store in SharePoint Portal.
- **Keywords.** Type in keywords that will enable improved queries in SharePoint Portal.
- **Description.** Type a description of the document that you are storing in SharePoint Portal.

- **Categories.** Add the **Categories** property to your document to enrich the set of properties that are stored with the document that are captured by search queries.
- **Check in.** Select the **Check in** check box if you want other users to open the file and make changes to it in SharePoint Portal. Until you check the file into SharePoint Portal, other users cannot check the file out. Only one copy of the file can be updated at one time. Users can also add comments to the file when checking the file in.
- **Publish.** Select this check box if you *do not* want the file to be viewed from a Web browser, although it is still stored in SharePoint Portal.
- **Check in Comments.** Before you check in the file, add comments about the changes you made to the file when you checked it out.
- **Rename.** Select this check box to rename the file.
- **Rename Schema.** Select this check box to rename the schema. You can then rename a file that is stored by using the SharePoint Portal v.1 component. If the file name that is being processed uses invalid characters (such as a "\", which can occur when using the PDF Barcode component), you must replace the invalid character with a valid character (such as "a").

Field Values tab

Use the options on this tab to add, modify, or remove field values.

The only difference between the SharePoint Portal v1.0 *Route* component and the SharePoint Portal v1.0*Process* component is that SharePoint Portal v1.0 *Process* component has two additional fields on the **Field Values** tab: **Field Passthrough** and **URL Field Name**. **Field Passthrough** and **URL Field Name** are not available for the Route component.

- **Add field values.** Click **Add field values** to add new field names and field values to the file that is being stored in the SharePoint Portal v1.0 component. You can add fields such as the title of the file, or any other fields that you require.

  When you click **Add field values**, the **Field Values** dialog box appears. Click the browse button (**…**) to open the **Select Field** dialog box. Select the fields that you want to add to your file. These fields can change, depending on which document library you select. Each document library has its own set of fields. The field types that are currently supported are Text, Number, Boolean, Currency, and Note (that is, multiple lines of text).
- **Modify.** Click **Modify** to modify the field value attributes.
- **Remove.** Click **Remove** to remove a field value.
- **Field Passthrough (for the Process component only).** Select this check box if you want the SharePoint Portal v1.0 Process component to pass all of the fields that it was unable to use on to the next component in the process.
- **URL Field Name (for the Process component only).** Type a URL Field Name string (SPSURL is the default). This string becomes the reserved replacement tag name (RRTN) for the URL. For example, using the default field name SPSURL, you can reference the URL as "**~SPS::SPSURL~**".

  If instead of SPSURL, you type the string "myurlvalue" for **URL Field Name**, the RRTN would be "**~SPS::myurlvalue~**".

Using Knowledge Package Loader to configure the SharePoint Portal v1.0 component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the SharePoint Portal v1.0 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the SharePoint Portal v1.0 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the SharePoint Portal v1.0 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the SharePoint Portal v1.0 component

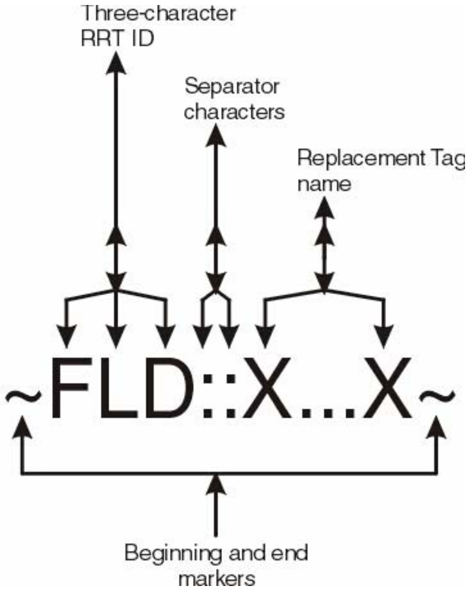
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the SharePoint Portal v1.0 component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and document time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **SPS**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	The original file-extension value.
URLFieldName	The URL of the file that is stored in the SharePoint Portal v1.0 Server (for the Process component only).

The following is an example of the RRTN process:

~SPS::FileName~~SPS::Counter~ is replaced with the filename that is created by using the SharePoint Portal v1.0 component.

~SPS::SPSURL~ is replaced with the URL of the file (where SPSURL is the field name designated for storing URLs). For the Process component only.

NOTE The RRTN values defined previously can only be used with the **Rename Schema** field of this component. You cannot use **~SPS::FileName~**, **~SPS::Counter~**, or **~SPS::FileExt~** in any other component except SharePoint Portal v1.0, and the RRTN must be used with the **Rename Schema** field of this component.

Field Replacement Tag Names

This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem description	Solution
The component cannot run.	Make sure that all of the necessary fields have been supplied. These include the server, workspace, user name, password, folder path, content source, and workflow fields.

Problem description	Solution
You cannot gain access to a folder path or a content source by using the browse buttons.	SharePoint Portal Server 2001 client Components (SPSCLIENT.MSI) are mandatory system requirements and must be installed before you can browse in **Folder Path** or **Content Source**. If SPSCLIENT.MSI is installed, make sure that you have typed in a valid user name, password, and server.

Restrictions and limitations

- Do not add two fields that have the same name to **Field Values**.
- The **URLFieldName** cannot have invalid characters such as a "\" or a "." (applies to the SharePoint Portal v1.0 Process component only).
- SharePoint Portal Server 2001 Client components, SPSCLIENT.MSI, must be installed on the AutoStore Server.

SharePoint Portal 2003 Process component

Use the SharePoint Portal 2003 component to store documents into a centralized, unified interface for enterprise users and highly flexible deployment options.

The only difference between the SharePoint Portal 2003 *Route* component and the SharePoint Portal 2003 *Process* component is that SharePoint Portal 2003 *Process* component has one additional field on the **General** tab: **Pass-through**. **Pass-through** is not available for the Route component.

In an AutoStore process, the SharePoint Portal 2003 component uses Microsoft SharePoint Portal Server 2003 for document management. Use this component to store files in Microsoft SharePoint Portal Server.

The most current version of the SharePoint Portal 2003 component does not require that you install .NET on the computer that is running the AutoStore software.

Feature highlights

You can perform the following tasks by using the SharePoint Portal 2003 component:

- Specify a location for storing files.
- Rename files that have duplicate names by using a schema name.
- Change document attributes.
- Repeat file names. The SharePoint Portal 2003 component appends duplicate filenames with a counter. For example, if the original file name was TEST.TXT, the component rename schema can rename the files to TEST1.TXT, TEST2.TXT, TEST3.TXT, and so on.

Using the SharePoint Portal Process component

In an AutoStore process, the SharePoint Portal 2003 component is frequently used with the Digital Sender component and the Poll Directory component. In this type of process, the SharePoint Portal 2003 component picks up files and stores them in Microsoft SharePoint Portal Server 2003.

For example, if a file is stored in a folder that you want to share with the rest of the company, you can create a process that uses Poll Directory as the Capture component. Save the file that you want to share in a designated Input folder, and then use SharePoint Portal 2003 as the Route component.

You can use the SharePoint Portal 2003 component as a Process component only when the Route component in the AutoStore process can accept field values (such as Lotus Notes, open database connectivity [ODBC], and so on).

NOTE To connect to a remote SharePoint Portal server from a client machine, you must run the SP2003WEBSERVICESETUP.MSI file that is located in the WebService Installation folder on the SharePoint server. To do this, locate the WebService Installation folder in the AutoStore directory on the client machine, copy it to the machine that is running SharePoint server, and then run the setup file.

Configuring the SharePoint Portal 2003 component

Depending on which Capture component you are using, follow the appropriate procedure to open the **SharePoint Portal 2003 Server** dialog box and configure the SharePoint Portal 2003 component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the SharePoint Portal 2003 component.

The following attributes are available in the **SharePoint Portal 2003 Server** dialog box.

General tab

Use the options on this tab to specify where you want to save incoming files.

The only difference between the SharePoint Portal 2003 *Route* component and the SharePoint Portal 2003 *Process* component is that SharePoint Portal 2003 *Process* component has one additional field on the **General** tab: **Pass-through**. **Pass-through** is not available for the Route component.

- **Server.** When you add the SharePoint Portal 2003 component, you must specify which server will store the documents.
- **User name.** Type a valid user name. If the user name that is entered is not an administrator on the SharePoint machine, SharePoint will not allow the user to log in.

NOTE

If the user name is not an administrator on the computer on which SharePoint Portal 2003 is installed, the SharePoint Portal 2003 component does not allow the user to log in.

- **Password.** Type a valid password.
- **Domain.** Type the name of the domain on which SharePoint Portal 2003 is running.
- **Site.** Use the Site directory to create various sites to store your documents. Select a site, and then select the document library where you want to work. If you do not specify a site, and then you decide to browse for a document library, the document libraries located in the base of the SharePoint Portal (root directory) are shown. Therefore, a site is not required, although a document library is required.
- **Document Library.** Select the document library where you want to work. Document libraries can have document libraries within them. Therefore, when you click the browse button (**...**) for **Document Library**, a tree view shows the document libraries hierarchy. You can create infinite document libraries inside document libraries if your computer has enough memory to support them.

  To enable the **OK** button, select a work site. You can store documents in only the work site, which is a subordinate document library. After you select the document library, you must then select the Folder Path for the document library.
- **Folder Path.** Select the folder path of the Document Library. Folders can have multiple folders within them. The **OK** button is available when you select a folder. A folder might already have other folders in it.

  You can create a folder dynamically by typing a new name for the folder.

  If a folder path is not specified, the file is stored in the root directory of the document library.
- **Rename.** Select this check box to rename the file.

- **Rename Schema.** Select this check box to rename the schema. You can then rename a file that is stored by using the SharePoint Portal 2003 component.

NOTE

If the **Rename Schema** field is left blank, this field is set to **~SPS::FileName~~SPS::Counter~~SPS::FileExt~**.

For example if the **Rename** check box is selected but the **Rename Schema** field is left blank, and the file "test.doc" was processed through this component, the rename schema will change the name of the file to "test1.doc".

- **Overwrite**. If you do not select the **Rename Schema** check box, this field determines whether a file that already exists in SharePoint Portal 2003 with the same name will be overwritten. If the **Overwrite** check box is not selected, the process will not store a file that has the same name as a file that already exists in the designated folder.
- **Pass-through.** Select this check box when you want the component to pass the document to the next component in the process. This is only valid for the SharePoint Portal 2003 eConnector (Process) component.

Columns tab

Use the options on this tab to add, modify, or remove field values.

- **Add...** Click **Add...** to add new field names and field values to the file that is being stored in the SharePoint Portal 2003 component. You can add fields such as the title of the file, or any other fields that you require.

  When you click **Add...**, the **Field Values** dialog box appears. Click the browse button (**…**) to open the **Select Field** dialog box. Select the fields that you want to add to your file. These fields can change, depending on which document library you select. Each document library has its own set of fields. All field types are supported.
- **Modify.** Click **Modify** to modify the field value attributes.
- **Remove.** Click **Remove** to remove a field value.

Using Knowledge Package Loader to configure the SharePoint Portal 2003 component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the SharePoint Portal 2003 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the SharePoint Portal 2003 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.

5. In the **Name** column, select the SharePoint Portal 2003 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the SharePoint Portal 2003 component

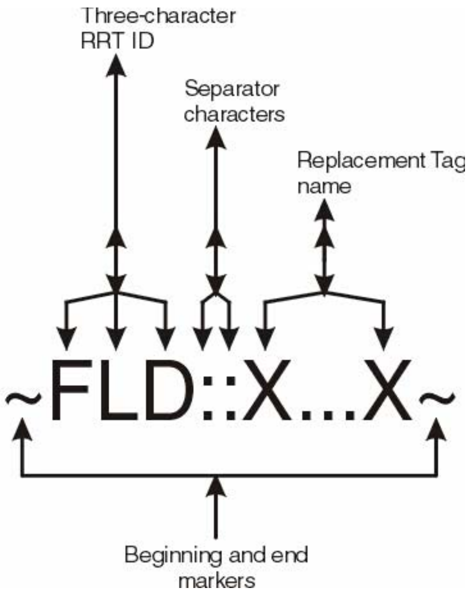
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the SharePoint Portal 2003 component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and document time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **SPS**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	The original file-extension value.
URL	The URL of the file that is stored in the SharePoint Portal server.

The following is an example of the RRTN process:

~SPS::FileName~~SPS::Counter~ is replaced with the filename that is created by using the SharePoint Portal 2003 component.

~SPS::URL~ is replaced with the URL of the file that is stored in Microsoft SharePoint Portal Server 2003.

NOTE

The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~SPS::FileName~**, **~SPS::Counter~**, or **~SPS::FileExt~** in any other component except the SharePoint Portal 2003 component, and it must be used with the **Rename** field.

This rule does not apply to the RRTN value, URL.

When using the SharePoint Portal 2003 Process component, do not use RRT **~SPS::URL~** in a subsequent component's rename schemas. This is because the value of this RRT contains backslashes and if you used it as part of a rename schema, it would create an error because file names cannot contain a backslash. For more information, see the Restrictions and limitations section.

Field Replacement Tag Names

This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem description	**Solution**
The component cannot run.	To connect to a remote SharePoint Portal server from a client machine, you must run the SP2003WEBSERVICESETUP.MSI file that is located in the WebService Installation folder on the SharePoint server. To do this, locate the WebService Installation folder in the AutoStore directory on the client machine, copy it to the machine that is running SharePoint server, and then run the setup file.
You are unable to locate documents that you stored in the Forms folder.	Avoid storing documents into the Forms folder. Instead, create a new folder and store documents into that newly created folder.

Restrictions and limitations

- Avoid storing documents into the Forms folder. Instead, create a new folder and store documents into that newly created folder.
- For Currency, only valid numbers are acceptable. Do not use **$** or other characters.
- If the **Rename Schema** field is left blank, this field is set to **~SPS::FileName~~SPS::Counter~~SPS::FileExt~**.

- If the user name is not an administrator on the computer on which SharePoint Portal 2003 is installed, the SharePoint Portal 2003 component does not allow the user to log in.
- When using the SharePoint Portal 2003 Process component, do not use RRT **~SPS::URL~** in a subsequent component's rename schemas. For example if your process contains the SharePoint Portal 2003 Process component and a Folder Store Route component, you cannot use the **~SPS::URL~** in the **Rename Schema** field of Folder Store component. This is because the value of this RRT value contains backslashes, and if you chose to use it as part of a rename schema, it would create an error because file names cannot contain backslashes.

OmniPage OCR component

Use the OmniPage OCR to perform optical character recognition (OCR) on images. This component provides image pre-processing, recognition, and multiple output formats as well as Zone-OCR capability.

In addition to recognition, this component offers color, grayscale, and black-and-white image input, support for multiple-image file formats, and image enhancement technologies like deskewing, auto-orientation, and intelligent page-layout decomposition, which help provide great recognition accuracy.

The OmniPage OCR component can deliver precise data for each recognized character, giving your process a lot of control over the output text format (at one extreme producing mirroring of the input document, and at the other permitting a unique, user-defined style). This component also provides five PDF converters to produce different types of .PDF file outputs including image-only, searchable, and other types.

It is recommended that you install Microsoft Windows 98, Millennium (Me), 2000, or XP on the server to achieve the best performance.

Feature highlights

You can perform the following tasks by using the OmniPage OCR component:

- Select the language.
- Select the code page.
- Output the entire OCR Text as either a file (specify the format), or to a field by using RRT.
- Use Zone OCR to draw the zones on the image, to select the filter type, and to name the image file.
- Choose the accuracy of recognition.
- Create a user dictionary, in which you can add your own words or regular expressions to the dictionary to increase accuracy.
- Select an image-inversion scheme before processing.
- Select image rotation before processing.
- Select the resolution enhancement before processing.
- Select the deskew and despeckle option for the image before processing.
- Select the format retention (save all the format, save only character and paragraph formatting, or save only the font size and name).
- Specify the paragraph formatting as either line spacing or alignment.
- Specify the character font and page formatting.

Using the OmniPage OCR component

This component can process any input type, such as .TIF or .BMP files. Use the OmniPage OCR component to perform the following tasks.

- **Zoned OCR capability.** Use the OmniPage OCR component to create zone OCR templates and apply these templates to the images. This component also provides the capability to apply filters on captured zoned fields.
- **Output formatting.** The component formats the output to most popular file formats, such as .PDF, .RTF, .TXT, and .XLS.
- **Dictionary.** Use the dictionary to assist in recognition and correction of OCR results.
- **Multi-Language Support.** This component supports multiple language OCR.
- **Dynamic use of OCR text.** Consider a process in which the Capture component is an MFP, followed by the Data Interchange Process component, OmniPage OCR component, and the Knowledge Package Builder Route component. In the MFP Capture component, you can set the properties of the Knowledge Package Builder **General** tab, and choose to Embed Documents and to Include Field Values. On the **Field Values** tab, you can insert two fields. You might name the first field **OCR Content**. Its value should be **~SSO::OCRText~**. You might name the second field **ZoneField**. Its value should be **~SSO::%zone1%~**. You can then run this as a service to produce an .XML file that has these two fields and the field values replaced with the text of the entire document for OCR content, and the text that was manipulated with OCR in the zone that you drew for zone1.

Configuring the OmniPage OCR component

Depending on which Capture component you are using, follow the appropriate procedure to open the **OmniPage OCR Configuration** dialog box and configure the OmniPage OCR component.

General tab

The **General** tab contains the following options:

- **Activate.** Select this check box to active the OmniPage OCR processing.
- **Delete original message.** Do not include the original image when the output is to the field, file, or zone fields.
- **Language.** Select the language in which you would like to have the recognition based.
- **Code.** The current Code Page setting should be able to express all characters that are validated for recognition (for example, the Character Set). Conflicts can occur between the set of characters validated for recognition (for example, the Character Set) and the Code Page selection. A selected Code Page might not support some characters. For example, if you select the Hungarian language and the current Code Page is Windows ANSI (Code Page 1252), the final output document will not contain some accented characters for that language.

- **Output Entire OCR Text as.** This item has three options for the output format of the document. You can output the entire OCR text as a File, Field, or both.
  - **File:** This option will output the final output to an actual document, rather than a field. (For example, if the file placed in the Poll directory were named TEST.JPG, and you wanted to process it to a .PDF file, the new file name would be TEST.PDF.) The image is Auto-zoned, meaning that the image is parsed and analyzes the structure of the page layout of the image and locates blocks for further processing, without specifying the zones manually.
  - **Field:** This option outputs to a field. The RRT "**~SSO::OCRText~**" references the value of this field. Auto-zoning is used here, in order to save all of the text in the image to the field as well.
- **Zone OCR.** If you want to add a zone to a page of the image, select this check box and click the **Configure** button. An **Open File** dialog box appears. In the lower-right corner, type the page number in which you want to place a zone.

  Zone fields will not be processed if the zone is not text. Images cannot be included into zone fields. When using zone fields, the component creates .ANN files for each page of the template file that you open and add a zone to. To run the component successfully, these files cannot be deleted.

  Click **Open** to open the zone OCR dialog box, where you can draw the zones that you want to save to a field. For each zone that you create, the name of that zone is used in field replacement. The value of the field will be the text within that zone. These zones should have a name and a filter type. Only one page at a time can appear.

  When you draw a field, you are prompted to give the field a unique name. This field name is used in field replacement. Whatever is inside the contents of this zone will be the value of the field.

  The following are the available filter types:
  - **Digits** recognizes numbers.
  - **Uppercase** recognizes uppercase letters, including accented ones.
  - **Lowercase** recognizes lowercase letters, including accented ones.
  - **Punctuation** recognizes punctuation such as "!" and "?".
  - **Miscellaneous** recognizes miscellaneous characters such as "+".
  - **Default** indicates that all other filter types are available and turned on.
- **Trade-off Setting.** You can select the Trade-off setting during recognition at page level. This setting has a trade-off influence between the speed and accuracy. The most accurate recognition will result in the slowest processing speed. This setting might also influence which auto-zoning algorithm will be applied, which will increase or decrease the speed and the accuracy. The following three options are available:
  - **Most accurate processing** (slowest)
  - **Mid-level accuracy/speed processing**
  - **Least accurate processing** (fastest)

Format tab

This tab has two options: Format retention and Settings.

Format retention

Select from the following three option settings to define the level of the format retention in the final output document:

- **Retain All format.** This option retains all formatting, including text zoning (the default setting). It keeps the original layout of the pages, including columns. This can include text, graphical and table zones, and frames. This is offered only for target programs that can handle these components. It is the only choice for all .PDF export, except for the file type "Adobe PDF edited."
- **Retain character and paragraph formatting.** This option retains a partial set of output formatting elements. All recognized paragraph and character formatting features, along with graphics and tables, are exported to the final document.
- **Retain font size and name.** This option retains only font size and font name formatting, as applicable. When exporting to .TXT file types, graphics and tables are not supported.

Settings tab

This tab covers the parameter settings for the following items:

- **Paragraph tab.** The **Paragraph** tab has two options: Line-spacing and Alignment.
  - **Line-spacing.** Use this option to define the line spacing of the output file. You can select **Auto-export** (the line spacing is calculated automatically), **Do not export** (the line spacing is set to single space), or **Pre-define** (double-line spacing, half-line spacing, one-and-a-half line spacing, or single-line spacing).
  - **Alignment.** Use this option to define the alignment of the output file. You can select **Auto-export** (the alignment is calculated automatically), **Do not export** (the alignment is set to default), or **Pre-define** (centered, justified (default), left-aligned, or right-aligned).
- **Character tab.** The **Character** tab has the following options:
  - **Auto Export.** Select the font attributes from the drop-down list for each of the attributes.
  - **Do not export.** Select this option if you do not want to select any specific font options.
  - **Pre-define.** Pre-define the font name and font size for the output document.
  - **Retain Underline.** Select the appropriate check box to retain the following document properties in the final output: Underline, Italic, or Bold.
- **Page tab.** The **Page** tab has the following options:
  - **Paper Size.** Define the output paper size as **Auto-export**, **Do not export**, or **Pre-define**. Use **Pre-define** to select either **Orientation** (Landscape or Portrait), or the size of the image (for example, Letter or Executive).
- **Retain Page Breaks.** Use this option to keep the page breaks in the final output document.

- **Graphics.** Use this option to select the level of graphics conversion for the image. The available levels are **Convert to 24-bit color**, **Convert to 8-bit grayscale**, **Convert to B/W**, **Suppress all graphics**, and **Transmit without conversion**.
- **Table Data.** Use this option to define how tables should appear (either by row-column format or tab format).

Pre-process Image tab

This tab includes several options for pre-processing images.

Inversion

Image inversion can be done on black-and-white, grayscale, or color images. For recognition, the feature requires black or dark characters on a white or pale background. The inverted image replaces the original one and becomes available to the program. Select the type of document-level inversion from the following options:

- **Automatic.** Each image is examined and is inverted if necessary. With this option, grayscale or color images undergo an implicit secondary conversion to create a clean, black-and-white image.
- **No images are inverted.**
- **All images are inverted.**

If **Automatic** Inversion and **Despeckle** are selected, a black-and-white image with over 280 dpi, a despeckled black-and-white image is created. Detecting whether inversion is needed or not, is done on the black-and-white image – the despeckled one if it is available. These black-and-white images are not available to the program.

Rotation and Mirroring

Select the appropriate image rotation from the drop-down list. Image rotation in multiples of 90° can be applied to black-and-white, grayscale, or color images. The following options are available:

- Rotate 180°
- Rotate left 90°
- Rotate right 90°
- Images are rotated automatically if necessary
- Mirrored around Y-axis
- Mirror, then rotate left 90°
- Mirror, then rotate right 90°
- Mirror, then rotate 180°
- Rotation is not performed

Resolution enhancement

Resolution enhancement is performed internally during image conversions. This doubles the resolution of a black-and-white image after an implicit secondary image conversion from grayscale or color. Select the resolution enhancement that you want from the following options:

- Enhance all images with 150 dpi or less (default)
- Forces resolution doubling in all cases
- Prohibit enhancement

Deskewing

You can perform image deskewing on black-and-white, grayscale, or color images, either automatically or with a configured value. A deskewed image should increase auto-zoning and recognition accuracy. The following options are available:

- Skew up to 15° is auto-detected and removed (default).
- No images are deskewed.
- Deskew all images by a configured number of pixels. (All images can be deskewed up to 30°.)

Despeckling

Image despeckling only relates to black-and-white images. The image can be imported as black-and-white or can be generated, explicitly or implicitly, from grayscale or color images. Image despeckling can improve the quality of auto-zoning, recognition, and the automatic pre-processing transformations.

The detection of inverted characters, skew, and orientation is more reliable on a despeckled black-and-white image. If you want the despeckled image to replace the original document, select the check box on the **Pre-process Image** tab.

User Dictionary tab

This tab offers several options for configuring the user dictionary.

- **Add.** To add entries to the user dictionary, click **Add**. In the dialog box that appears, type the value of the field, along with its type. The valid value types are **Regular expression** and **String**.
- **Modify.** To modify the value or type of an entry in the user dictionary, highlight the item and click **Modify**.
- **Remove.** To remove an entry in the user dictionary, highlight the item and click **Remove**.
- **File Name.** Select path and name of the dictionary file that you want to save to. This file will be used as the user dictionary in your process.

- **Enable Spell Check.** Select this check box to turn on spell checking. Selecting this option can improve accuracy by automatically correcting misspelled words.
- **Enable the correction of non-compliant words.** If this option is turned off, non-compliant and questionable words are flagged, but are not corrected automatically. When you turn on this feature, the non-compliant words are changed and are marked as changed.

Using Knowledge Package Loader to configure the OmniPage OCR component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the OmniPage OCR component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the OmniPage OCR component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the OmniPage OCR component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the OmniPage OCR component

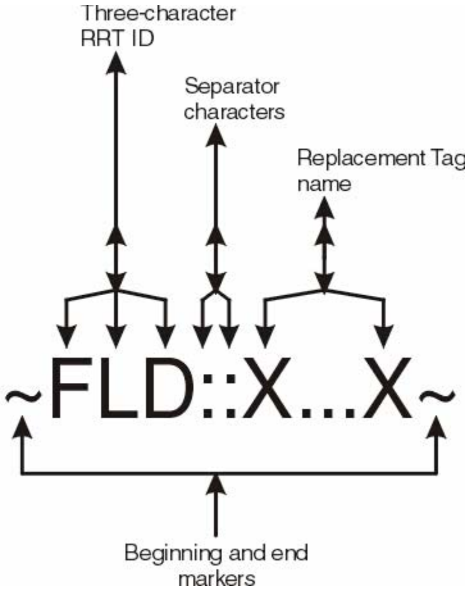
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the OmniPage OCR component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **SSO**.

Reserved Replacement Tag Names (RRTN)

~SSO::OCRText~ references the OCR text of the document.

Field Replacement Tag Names

This is an example of a Field Replacement Tag Name (FRTN):

~SSO::%abc%~ references the corresponding Zone OCR field that is named abc.

Special Set Replacement Tag Names (SSRTN)

This component does not support any SSRTNs.

Troubleshooting tips

Problem description	Solution
The OCR function failed in the specified zone.	The zone field type or name was not given. If its name was given, it should be unique. Another possibility is that the zone that was selected did not contain any recognizable text.

Restrictions and limitations

- This component creates .PDF Version 1.3 files (by using Acrobat 4.*x*).
- The best way to prevent multiple zones from having the same name is to distinguish them according to the page number (for example, "**p1:zone1**").
- Overlapping zones can cause errors in your process.

Basic Image Management component

Use the Basic Image Management component to clean up black-and-white multipage images. You can also use this component to split a document into multiple pages and provide barcode recognition. You can combine these functions according to your business needs.

The Basic Image Management component uses LeadTools Technologies to handle image files and detect barcodes.

Feature highlights

The Basic Image Management component includes the following features:

- Improves the quality of black-and-white .TIF images by reducing skew (deskewing) and removing speckles (despeckling).
- Reads eight types of barcodes.
- Uses coordinates to search barcodes by zone.
- Splits images by using either of two methods: by specifying the number of pages, or by matching the barcode to a specified value or pattern.

The values from the barcode can be used as parameters in subsequent components in the configuration. They can also be mapped into fields that will automatically be used by Capture or Process components that can use this field information.

This component is commonly used to read barcodes that appear in black-and-white .TIF files that a scanning device has generated. The image must have at least a 200-dpi resolution. Depending on the scanner quality, and the size and type of the barcode, you might need to generate higher-quality images in order for the barcodes to be read correctly.

If you want to set up your process so that other components can use the barcode values, configure this component through a blocking component, such as Digital Sender, MFP, or POP3 email. If you plan to use this component with a non-blocking Capture component, place a Data Interchange process component before the Basic Image Management component.

If barcodes are not being read, or if they are only used for splitting pages, the component does not have to be configured through a blocking component.

Using the Basic Image Management component

This component is used as part of business processes that rely on barcode coversheets or barcode information inside a document (for example, where a barcode is used to represent the invoice number in an invoice document). This component can be used to read the values of a barcode and then store such information together with the document in one of the supported document-management systems.

The information can also be used to dynamically set other values in the configuration. For example, if the invoice is being sent to a Folder Store component, the final file could be renamed by using the invoice number as part of the new name.

The Basic Image Management component is commonly used where the quality of the final image is essential. Use deskewing and despeckling functions to improve the quality of the scanned document.

If you have a process that receives an image file, and that contains several documents that have an equal number of pages each of which has to be stored separately, you can use the Split function on a specific number of pages to implement batch processing and increase efficiency.

Configuring the Basic Image Management component

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Basic Image Management component.

The following attributes are available in the Basic Image Management configuration dialog box:

Image Processing tab

The **Image Processing** tab offers the following features:

- **Image Processing.** Select this check box to activate the image clean-up and image-splitting functions, and to change the component configuration.
- **Deskew.** Select this check box to deskew (straighten) the image.
- **Despeckle.** Select this check box to despeckle (remove small marks from) the image.
- **Split Type.** Split Type offers the following options:
  - **Split Into.** Select this check box to split the document into the number of pages that you specify in the edit box.
  - **Split on Barcode Value.** Select this check box to split the document every time that the component detects the barcode that is specified in the edit box. The barcode value can contain wildcards such as ***** and **?**. For example, if "*" is specified in the edit box, then the document splits when it finds any barcode. If **1000?** is specified, then the document splits if the barcode is **10002**, **10003**, and so on.
  - **Keep Barcode Page.** If the split is based on the barcode, you can decide to either keep the barcode page or discard it.
  - **Pre separator.** Select this check box when you are using the barcode page as a separator and you want the barcode to serve as a separator at the beginning of the document. If this check box is not selected, the component assumes that the barcode separator is the last page of the split document and starts a new file after it.

Barcode Tab

Use this tab to specify the barcode attributes.

- **Activate.** Select this check box to turn on the barcode processing function.
- **Barcode Type.** If you know the barcode type that is going to be read, select it. If not, select **All linear**, and the component will read the by barcode using any of the supported barcode types. The following are supported barcode types:
  - **Codabar.**
  - **Code 128.** A very high-density alphanumeric barcode that contains 106 different printed barcode patterns.
  - **Code 3 of 9 (Code 39).** An alphanumeric barcode that is simpler than Code 128.
  - **EAN 13.** European Article Numbering consisting of 13 numbers.
  - **EAN 8.** European Article Numbering consisting of eight numbers.
  - **Interleaved 2 of 5.** A numeric-only barcode that is used for interleaving pairs of numbers in a high-density barcode format.
  - **UPC Version A.** A Universal Product Code consisting of 12 numbers.
  - **UPC Version E.** A Universal Product Code consisting of eight numbers.
- **Max Barcode.** Specify a maximum number of barcodes that the component should try to read from a page. If you set this value to 0, the search engine looks for barcodes until the end of the page. Specifying a maximum number of barcodes can increase performance, because the component stops searching the page for barcodes after it has read the specified number of barcodes.
- **Search Zone.** Restrict the image search area manually by specifying the coordinates, either in inches or millimeters. Use the syntax in the text box:
  - **Top.** The distance from the upper side of the page to the beginning of the search area.
  - **Left.** The distance from the left side of the page to the beginning of the search area.
  - **Right.** The distance from the left side of the page to the end of the search area.
  - **Bottom.** The distance from the lower side of the page to the end of the search area.

  You can specify the search zone by graphically selecting the zone from a sample image file. To do this, click "**...**" and select a sample image file from which to identify a search zone, and then click **Open**. You will be presented with a preview of the first page of the image. Move the mouse to the preview area. The cursor will change appearance. Click on the image and drag the cursor to select a square section of the image. Click **OK**. This is the area where the component will search for barcodes. You can change the file used for previewing by clicking **Open** and selecting a different image.
- **Field Values.** Click **Add** to provide the field name that specifies the barcode value that should be associated with this field. Click "**...**" and provide the page number and barcode value of the barcode that contains the information that should be assigned to this field.

  You can also specify the barcode value directly by using a specific syntax.

Using Knowledge Package Loader (Capture) to configure the Basic Image Management component

1. Double-click the Knowledge Package Loader Capture component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Basic Image Management component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Basic Image Management component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Basic Image Management component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Basic Image Management component

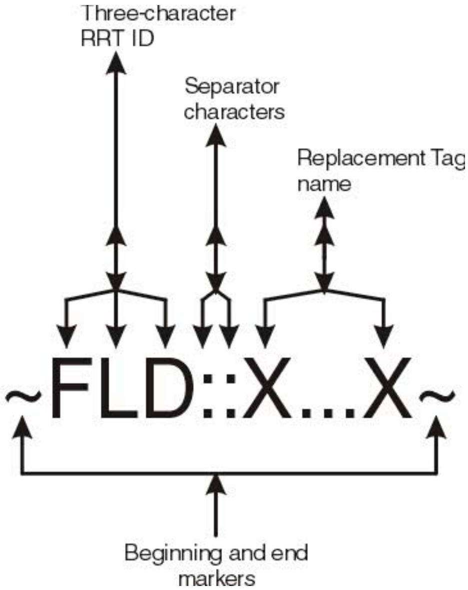
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Basic Image Management component in the **Component Name** window.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provides useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID of this component is **L1B**.

If barcode reading is activated, the values for single barcodes in different pages can be used to dynamically configure other components in the configuration. By using different variations in referencing the page number and location of the barcode, you can not only refer to a single value but also to the concatenation or several values or even all of the barcodes in a document.

The following is an example of the syntax that refers to barcode values that were read from the image file:

~L1B::PageNumber, BarcodePosition~

In this value, **PageNumber** is the image page where the barcode is located, and **BarcodePosition** is the position of the barcode on the page, counting from left to right and from top to bottom.

The following special characters can be used instead of a specific **PageNumber** or **BarcodePosition**:

- ***:**: Any Page Number or Barcode Position
- **-1**: Last Page or Barcode Position

The possible combinations of these values are shown in the following table.

~L1B::*, BarcodePosition~	This returns the concatenation of all barcode values in the **BarcodePosition** on any page of the image file.
~L1B::PageNumber,*~	This returns the concatenation of all barcode values for any position on the **PageNumber** page of the image file.
~L1B::-1,-1~	This returns the barcode value of the last barcode on the last page of the image file.
~L1B::-1, BarcodePosition~	This returns the barcode value in the **BarcodePosition** on the last page of the image file.
~L1B::*,*~	This returns the concatenation of all of the barcode values for any position on any page of the image file.

NOTE

If you are using splitting on barcode, the term last page refers to the last page of the image file after it has been split.

Troubleshooting tips

Problem description	Solution
The Basic Image Management component cannot read one of the supported barcode types.	The search zone does not include the area where the barcode is located. Using the image that contains the barcode as a sample, change the search zone so that it includes the location where the barcode appears. If you are not sure if this location is the same for all images, then do not use a search region. The component will try to find the barcode by reading the entire page.
The status monitor shows that the barcodes are being read, but the barcode values references are not being replaced.	The Capture component is not a Blocking component and you are not using the Data Interchange Process component in your process. Add the Data Interchange component to the process before the Basic Image Management component, and configure the remaining components through the Data Interchange component.

Restrictions and limitations

The Basic Image Management component can receive only 1-bit (black-and-white) .TIF images.

PDF 417 Barcode component

Use the PDF 417 Barcode component to read 2D barcodes from images. After a barcode has been read, other components can use the barcode information within the process. Use the PDF 417 barcodes when a large number of data fields need to be captured within a process.

Refero is an NSi Web server program that generates PDF 417 2D barcodes. The Refero program can produce indexing forms and link them to an open database connectivity (ODBC) database, a Microsoft SharePoint Portal Server, or an iManage document management system.

The PDF 417 Barcode component uses LeadTools Technologies to handle image files and detect barcodes.

Feature highlights

PDF 417 Barcode is used to read 2D barcodes of type PDF 417. This is a multirow barcode that can encode several hundred characters.

Subsequent components in the configuration can use the values from the barcode as parameters. These values can also be mapped into fields that the Route or Process components can use automatically (if the component uses this field information).

The component facilitates image splitting by instructing the component to split the document when a barcode matches a specific value or a pattern.

This component is commonly used to read barcodes that appear in black-and-white .TIF images that a scanning device has generated. The image must have at least a 200-dpi resolution. Depending on the scanner quality, and the size and type of the barcode, you might need to generate higher-quality images in order for the barcodes to be read correctly.

If you want to set up your process so that other components can use the barcode values, configure this component through a blocking component, such as Digital Sender, MFP, or POP3 E-mail. If you plan to use this component with a non-blocking Capture component, place a Data Interchange Process component before the PDF 417 Barcode component.

Using the PDF 417 Barcode component

This component is used as part of business processes that rely on barcode coversheets or barcode information inside a document (for example, when a barcode is used to represent the invoice number in an invoice document).

The PDF 417 Barcode component can be used to read the values of a barcode and then the information together with the document in one of the supported document-management systems.

The information can also be used to dynamically set other values in the configuration. For example, if the invoice is being sent to a Folder Store component, the final file could be renamed by using the invoice number as part of the new name.

Configuring the PDF 417 Barcode component

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the PDF 417 Barcode component.

The following attributes are available in the **PDF 417 Barcode** configuration dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Activate.** Select this check box to turn on the barcode processing function. Use this check box to turn on or turn off this component based on input or Refero barcode routing slips.
- **Max Barcode.** Specify a maximum number of barcodes that the component should try to read from a page. If you set this value to 0, the search engine looks for barcodes until the end of the page. Specifying a maximum number of barcodes can increase performance because the component stops searching the page for barcodes after it has read the maximum specified number of barcodes.
- **Barcode Orientation.** Use this option to specify the reading direction of the individual barcodes. The scanning direction is always top to bottom. This orientation refers to the barcode itself; however, note that the orientation can be changed by the way the page is scanned. Select the barcode orientation as follows:
  - **Autodetect.** Detect the barcode orientation automatically. Use this when the image scan orientation can change, and the barcodes can appear upside down.

NOTE

The following four barcode-orientation options allow faster processing because no orientation detection is required and all barcodes are assumed to be uniformly aligned.

  - **Top to Bottom.**
  - **Bottom to Top.**
  - **Right to Left.**
  - **Left to Right.**

As mentioned previously, the orientation can be changed by the way the page is scanned. For example: a page has one barcode oriented from Left to Right; and the page is scanned with a 90° orientation change toward the left; and the barcode orientation is set as Left to Right. This component cannot detect the barcode because now the barcode has a bottom to top orientation due to the 90° orientation change.

- **Search Zone.** Restrict the image search area manually by specifying the coordinates, either in inches or millimeters. Use the following syntax in the text box:
  - **Top.** The distance from the top of the page to the beginning of the search area.
  - **Left.** The distance from the left side of the page to the beginning of the search area.
  - **Right.** The distance from the left side of the page to the end of the search area.
  - **Bottom.** The distance from the bottom end of the page to the end of the search area.

  You can specify the search zone by graphically selecting the zone from a sample image file. To do this, click "**…**" and select a sample image file from which to identify a search zone, and then click **Open**. You will be presented with a preview of the first page of the image. Move the mouse to the preview area. The cursor will change appearance. Click on the image and drag the cursor to select a square section of the image. Click **OK**. This is the area where the component will search for barcodes. You can change the file that you use for previewing by clicking **Open** and selecting a different image.
- **Page Split.**
  - **Split on Barcode Value.** Select this check box to split the document every time that the component detects the barcode that is specified in the edit box. The barcode value can contain wildcards such as ***** and **?**. For example, if "*" is specified in the edit box, then the document splits when it finds any barcode. If **1000?** is specified, then the document splits if the barcode is **10002**, **10003**, and so on.

    By definition, the following wildcard characters can be used:

    ***** - Any value of any length

    ? - Any single character
  - **Keep Barcode Page.** If the split is based on the barcode, you can decide to either keep the barcode page or discard it.
  - **Pre Separator.** Select this check box when you are using the barcode page as a separator and you want the barcode to serve as a separator at the beginning of the document. If this check box is not selected, the component assumes that the barcode separator is the last page of the split document and starts a new file after it.
- **Barcode Data.**
- **Use Refero XML Schema.** Select this check box if the PDF 417 Barcode component that is used in the process is being generated from the NSi Refero program. You can instruct the component to automatically extract individual fields of information from the barcode. The barcode will still be available as a whole, and references to individual fields contained in it can be used.

Field Values tab

Use this tab to delete field entries from the list of field values.

- **Insert.** Click this button to type the field name and the barcode value that is associated with this field. Click "**...**" to provide the page number and barcode number of the barcode that contains the information that should be assigned to this field. You can also use RRT strings to dynamically specify the barcode value.
- **Delete.** Click this button to remove a field value entry from the field value list.

Use the appropriate procedure to open the **PDF 417 Barcode** configuration dialog box to configure the PDF 417 Barcode component.

Using Knowledge Package Loader (Capture) to configure the PDF 417 Barcode component

1. Double-click the Knowledge Package Loader Capture component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the PDF 417 Barcode component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the PDF 417 Barcode component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the PDF 417 Barcode component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the PDF 417 Barcode component

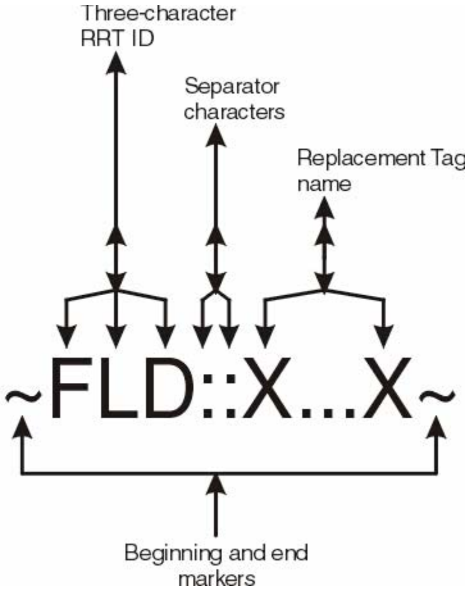
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the PDF 417 Barcode component in the **Component Name** window.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

If barcode reading is activated, the values for single barcodes on different pages can be used to dynamically configure other components in the configuration. By using different variations in referencing the page number and location of the barcode, you cannot only refer to a single value but also to the concatenation of several values or even all of the barcodes in a document.

The following syntax is used to refer to barcode values that were read from the image file:

~L2B::PageNumber, BarcodePosition~

where **PageNumber** is the image page where the barcode is located and **BarcodePosition** is the position of the barcode on the page (counting from left to right and from top to bottom).

Certain special characters can be used instead of a specific **PageNumber** or **BarcodePosition**:

***:**: Any page number or barcode position.

-1: Last page or barcode position.

The possible combinations are shown in the following table.

~L2B::*, BarcodePosition~	This returns the concatenation of all barcode values in **BarcodePosition** found on any page of the image file.
~L2B::PageNumber,*~	This returns the concatenation of all barcode values for any position found on the **PageNumber** page of the image file.
~L2B::-1,-1~	This returns the barcode value of the last barcode on the last page of the image file.
~L2B::-1, BarcodePosition~	This returns the barcode value in **BarcodePosition** found on the last page of the image file.
~L2B::*,*~	This returns the concatenation of all barcode values for any position found on any page of the image file.

NOTE If you are using splitting on barcode, then the term "last page" refers to the last page of an image file after it has been split.

Troubleshooting tips

Problem description	Solution
The PDF 417 Barcode component cannot read one of the supported barcode types.	The search zone does not include the area where the barcode is located. Using the image that contains the barcode as a sample, change the search zone so that it includes the location where the barcode appears. If you are not sure if this location is always going to be the same for all images, then do not use a search region. The component will look for the barcode on the entire page.
The status monitor shows that the barcodes are being read but the barcode values references are not being replaced.	The Capture component is not a blocking component and you are not using a Data Interchange Process component in your configuration. Insert the Data Interchange component in your configuration before the PDF 417 Barcode component, and configure the remaining components through the Data Interchange component.

Restrictions and limitations

- The splitting functionality is only available when reading multipage .TIF images.
- The following is a complete list of the file formats that the PDF 417 Barcode component supports.

JPEG formats

- JPEG File Interchange Format.
- Tagged Image File with JPEG compression.
- JPEG 2000 Format. This file format contains image data and extra information about the contents and organization of the file.

GIF formats

- CompuServe GIF.

TIFF formats

- Tagged Image File Format, with no compression and with RGB color space and 8-bit grayscale.
- Tagged Image File, with no compression and with CMYK color space.
- Tagged Image File, with no compression and with YCbCr color space.

- Tagged Image File with PackBits Compression and RGB color space.
- Tagged Image File with PackBits Compression and CMYK color space.
- Tagged Image File with PackBits Compression and color YCbCr space.
- Tagged Image File with CMP Compression.
- Tagged Image File with JBIG Compression.
- Tagged Image File with a vector image saved as a DXF.
- Tagged Image File with JPEG 2000 Compression. This file format contains only a stream of image data.
- Tagged Image File with Wavelet CMP Compression.

BMP formats

- Windows BMP, with no compression.
- Windows BMP, with RLE compression.
- OS/2 BMP version 1.*x*.
- OS/2 BMP version 2.*x*.
- Wireless Bitmap file. Type 0.

WMF and EMF formats

- Windows MetaFile.
- Windows Enhanced MetaFile.

Exif formats

- Exif file containing a TIFF image, with no compression and with RGB color space.
- Exif file containing a TIFF image, with no compression and with YCbCr color space.
- Exif file containing a JPEG compressed image.

1-Bit FAX formats

- TIFF, compressed using CCITT.
- TIFF, compressed using CCITT, group 3, 1 dimension.
- TIFF, compressed using CCITT, group 3, 2 dimensions.
- TIFF, compressed using CCITT, group 4.
- Raw FAX, compressed using CCITT group 3, 1 dimension.
- Raw FAX, compressed using CCITT group 3, 2 dimensions.
- Raw FAX, compressed using CCITT group 4.
- IOCA, compressed using CCITT group 3, 1 dimension.
- IOCA, compressed using CCITT group 3, 2 dimensions.

- IOCA, compressed using CCITT group 4.
- IOCA, compressed using IBM MMR, with the MO:DCA wrapper.
- IOCA, uncompressed, with the MO:DCA wrapper.

Other 1-Bit formats

- MacPaint.
- Portable Bitmap - ASCII File.
- Portable Bitmap - Binary File.
- XBitmap File.
- Microsoft Paint.

ABBYY FormReader v6.0 component

Use the ABBYY FormReader 6.0 component to extract information from printed forms and export it to databases and information systems that use ABBYY FormReader technology.

ABBYY FormReader 6.0 software must be installed on the server before you use this component. The ABBYY FormReader installation and licensing must be completed independently from the AutoStore system setup. ABBYY FormReader software is available from ABBYY Software House from which you can obtain the appropriate licenses and user documentation.

This section provides a brief description of the interface between AutoStore and ABBYY FormReader and how to use the results of Form Recognition to configure other components. See the documentation that came with ABBYY FormReader software for more information about that product.

Feature highlights

The following is the list of features that ABBYY FormReader 6.0 technology offers, all of which are available when using this component.

Scanning

- Image pre-processing
- Deskew
- Noise cleaning
- Page-orientation detection
- TWAIN scanner support
- Graphic format import (.TIF, .JPG, .PCX, .DCX, .PNG, .BMP)
- Manual input feeding
- ADF support
- Compensation of linear distortions using cornerstones (for example, faxed forms)

Recognition

- OCR/ICR Fountain Image Transformation Technology
- Handprint
- Machine print
- Mixed text
- Mark sense (OMR): check boxes, radio groups, any type of marking
- Barcode recognition: EAN13, EAN8, Check Code 3, Check Interleaved 25, Code 39, Code 128

- Handprint support for 16 languages: English, German, French, Italian, Russian, Polish, Ukrainian, Bulgarian, Czech, Slovak, Lithuanian, Spanish, Dutch, Finnish, Romanian, and Turkish
- Bordered text: framed, text over comb, underlined
- Multi-line text
- Local detection of field position
- Built-in template designer
- Template auto-identification
- Various reference blocks: cornerstones, text, lines, image
- Fuzzy-logic interpretation
- Handwriting support: Europe, American, Russian, Japan, and Thai
- Scan and Read/Open and Read modes
- Background recognition
- Interpretation of check boxes that have been marked by mistake (shaded area)

Configuring the ABBYY FormReader v6.0 component

Set up the ABBYY FormReader 6.0 component attributes to achieve your business-process objectives, and then add a Route component. The image files are automatically converted to the required output format and delivered to the Route component.

Use static or dynamic values as defined in the Source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the ABBYY FormReader 6.0 component.

The following attributes are available in the ABBYY FormReader dialog box.

Batch tab

Use the options on this tab to set the following attributes:

- **Path.** Type the path to the batch definition file that will be used to process the scanned forms. You can either select an existing batch by clicking **Open Batch** and browsing for the appropriate batch, or you can create a new batch by clicking **Create Batch** to start the ABBYY FormReader batch Designer Wizard. When the wizard starts, follow the onscreen instructions to create a new batch.
- **Accuracy Threshold.** This parameter indicates the minimum acceptable accuracy when reading the form. It is specified in percentage, and the default value is 85%. This means that if at least 85% of the elements in the form are found, then the form will proceed; otherwise, the form is rejected.
- **Reject Images.**
- **Reject Images With Errors.** Select this check box to reject images that have errors. For example, you might want to reject images that have unexpected elements in any areas.
- **Reject Images With Warnings.** Select this check box to reject images that generate warnings.

- **Bad Accuracy Image Path.** Specify the folder path for images that did not meet the minimum accuracy that was specified in the accuracy threshold parameter.
- **Template List.** This is the list of templates that are contained in the batch. You can add, modify, or delete templates from the batch directly from the AutoStore interface. You can also add or modify the data types that are associated with the template.

Image tab

Use the options on this tab to set the following attributes:

- **Image Orientation.**
- **Rotate image 90° and 270°.** Select this check box to turn on automatic detection of the orientation of the image and rotate it 90° or 270°, if necessary, before performing the form recognition. If you do not select either of the **Rotate image** check boxes, the component attempts to recognize the images as they were originally scanned, regardless of the orientation.
- **Rotate image 180°.** Select this check box to turn on automatic detection of the orientation of the image and rotate it 180°, if necessary, before performing the form recognition. If you do not select either of the **Rotate image** check boxes, the component attempts to recognize the images as they were originally scanned, regardless of the orientation.
- **Image Filtering.**
- **Clean Images on Opening.** Select this check box if you want the image to be deskewed and despeckled before the form recognition starts.
- **Image Storing Parameters.**
- **Store Black and White Images.** Select this check box to change color and grayscale images to black-and-white format before form recognition starts.
- **Image Postprocessing.**
- **Delete Images.** Select this check box if you want the image deleted after the form recognition is complete. You can select this option when you only want to retain the data that is extracted from the form, rather than retaining the form itself.
- **Handprinted Text.**
- **Detect Handprint Text Style by Country/Region.** Select this check box if you want the handprinted text style to be selected automatically based on a geographic country/region.
- **Handprinted Text Style.** If you did not select **Detect Handprint Text Style by Country/Region**, then you must select one of the available handprinted styles. If you are not sure which one to select, select the one that uses an alphabet that is most similar to yours.
- **Advanced Options.** This button is for advanced users who are familiar with image processing technology. If you are not sure how to set these values appropriately, you should use the defaults that are provided, because inadequate values decrease the form-recognition success.

Export tab

Use this tab to provide information about exporting the data that is extracted from the image.

- **Save output as.** Provide information about how to export the data into two generic options:
  - **File.** If you select this option, the information can be saved to a specially formatted file. The possible file types are .DBF, .TXT, .DOS TXT, .CSV, .DOS CSV, .XML, and .XLS.
  - **Export All Fields.** If you select this option, all of the fields are passed down to subsequent components in the process without using any RRTs.
- **TXT.** Save the results as a .TXT file or a .DOS TXT file. When you save the file as .DOS TXT, you can provide the code page for the file in the .TXT file section.
- **DBF.** Save the results in .DBF file format. If you select this option, you can provide the code page for the file in the .DBF section.
- **CSV.** Save the results as a .CSV file or as a .DOS CSV file. When you save the results as a .DOS version of a .CSV file, you can provide the code page for the file in the .CSV section.
- **XML.** Save the results in .XML file format. If you select this option, you can specify that encoding be used in the output .XML file.
- **XLS.** Save the results in .XLS file format.

Use the appropriate procedure to open the **ABBYY FormReader** dialog box and configure the ABBYY FormReader 6.0 component.

Using Knowledge Package Loader (Capture) to configure the ABBYY FormReader 6.0 component

1. Double-click the Knowledge Package Loader Capture component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the ABBYY FormReader 6.0 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the ABBYY FormReader 6.0 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the ABBYY FormReader 6.0 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the ABBYY FormReader 6.0 component

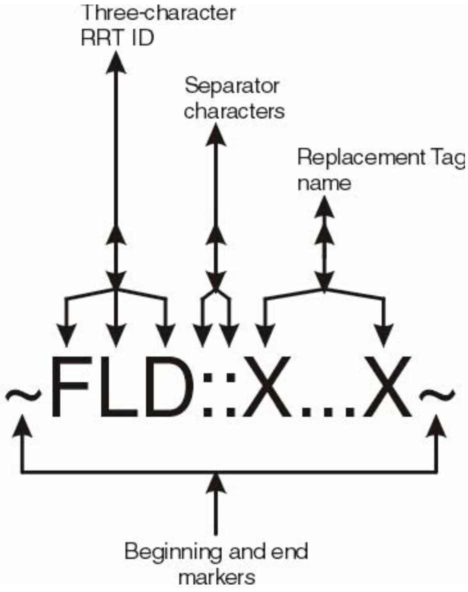
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the ABBYY FormReader 6.0 component in the **Component Name** window.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provides useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FR6**.

Use the following syntax to refer to fields that are extracted from images (FRTNs).

~FR6::FormNumber, FieldName~	Where **FormNumber** is the number of the recognized page and **FieldName** the name of the field that you want to reference. Note that the **FormNumber** does not necessarily correspond to the page number in the document. Blank pages in the document are skipped. You can use **"*"** instead of a static **FormNumber**. This indicates that you want to reference the first occurrence of **FieldName** without regard for **FormNumber**.
~FR6::MatchNumber, TemplateName,FieldName~	Where **TemplateName** is the name of the template that matched against the recognized page, **MatchNumber** is the number of times this template matched against a page (including the current one), and **FieldName** is the name of the field that you want to reference. For example, the **RTT ~FR6::2,MyTemplate,Age~ I** references the field "**Age**", which is matched from the template "**MyTemplate**" for the second form that is recognized with that template.

Troubleshooting tips

Problem description	Solution
The ABBYY FormReader hardware error message appears.	When you install the ABBYY FormReader component software, make sure that you select "Software Activation." If you select "Hardware Activation," uninstall and reinstall the ABBYY FormReader software so that you can select "Software Activation."

Restrictions and limitations

This component contains no known restrictions or limitations.

File Options component

Use the File Options component to capture the processed files into a directory and reroute them into multiple destinations. This component is useful when you want to eliminate expensive and repetitive CPU-intensive tasks. You can use the File Options component to determine how to treat a variety of files, depending on whether or not the files have been successfully captured.

"Process once and store many times" is the best description of the File Options component.

Feature highlights

The File Options component includes the following features.

- Depending on the file extension, you can either remove the files or place them into specified folders.
- Depending on whether the process was a success or a failure, you can define the output file option.

Using the File Options component

By using this component in a process, you can either remove processed files or place them in particular folders. This feature depends on whether the capture portion of the AutoStore process was a success or a failure. Use the File Options component to save and reuse the files that are created within a process to conserve time and resources that are required to distribute the same file out to other destinations. The File Options component increases accuracy through reuse of output files.

For example, you can use the Poll Directory, File Options, and Send to Database components to set up a folder for failures.

Using the File Options component, select **Move Files**. Then designate the path where you want to send failures. For example, specify the folder **C:\FAILURE**. If you do not already have the folder named **FAILURE**, you are prompted to create the directory. Click **Yes**, and AutoStore creates the folder **FAILURE** on your computer C:\ drive.

To use File Options as a Route component, select Send to Database, configure the parameters, and then run the process. If for any reason the files cannot be stored within the Send to Database component, then the files will be sent to the **FAILURE** folder. In this way, you can check for any failures when you have a large number of files in the directory that you specified in the Poll Directory component.

Configuring the File Options component

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the File Options component.

The following attributes are available in the **File Options** configuration dialog box.

General tab

Use the attributes in the **General** tab to define the file treatment.

When the process is successful:

- **Remove Files.** Delete the files when the process is successful.
- **Move Files.** Copies the processed files to the folder that you specify when the process is successful. If you select **Move Files**, you must specify a directory path where you are copying the files.

When the process fails:

- **Remove Files.** Delete the files when the process fails.
- **Move Files.** Copies the files to the folder that you specify when the process is a failure. If you select **Move Files**, you must specify you must specify a directory path where you are copying the files. Note that the files will not be removed from the source directory.

For general management:

- **Add.** Click this button to add a new tab that contains new success and failure options for the file extension that you specify. *Do not type the period preceding the file extension.* After processing, files that have the specified file extension (for example, .TXT, .PDF, .DOC) are moved to the path that you designate.
- **Remove.** Click this button to remove the active file-extension tab. You cannot remove the **General** tab.

Use the appropriate procedure to open the **File Options** configuration dialog box and configure the File Options component.

Using Knowledge Package Loader (Capture) to configure the File Options component

1. Double-click the Knowledge Package Loader Capture component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the File Options component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the File Options component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the File Options component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the File Options component

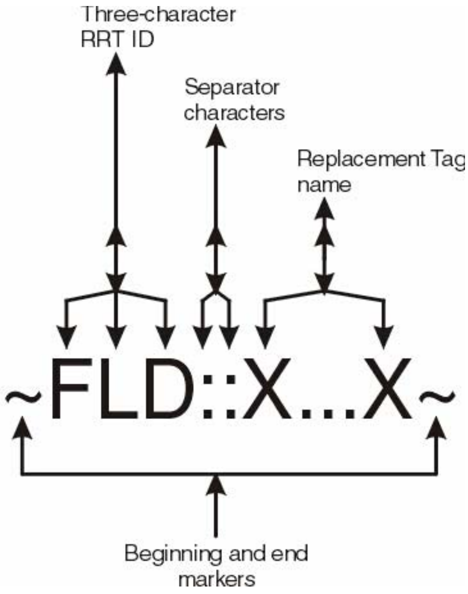
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the File Options component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The File Options component does not generate RRTs.

Troubleshooting tips

Problem description	Solution
A problem occurs when using Digital Sender with the File Options component.	If you place the File Options component directly after the Digital Sender component, then it overrides the Digital Sender options. If you have multiple process components, and you place a File Options component after your first process component, the success or failure of the first process component will not be affected by the file options. If the File Options component is in between two different process components, the second process component will be affected by the File Options component changes.
An error message appears when you select **Move Files** and create a new folder named SUCCESSFOLDER.	Do not specify a folder name with an entire directory pathname such as C:\SUCCESSFOLDER.
An error message appears (Error starting AutoStore service) when you try to start the Service Manager after adding file extensions.	Make sure that you only type in the letters of the file extension (for example, TIF, BMP, and so on), and not the preceding period.

Restrictions and limitations

- You must specify a folder when you select **Move Files**.

Send to FTP

Use the Send to FTP component to store files that are fed from a Capture component to the FTP site for storage. The Send to FTP component is available as either an eConnector (Process) or a Route component. This component provides additional control over the FTP site transmission by providing support for secured FTP sites and bandwidth flow-control.

You can design your process to accommodate your business process. Any client (anonymous or specific) can then open an FTP session to this server to retrieve documents and accomplish other necessary tasks in the business process.

The Send to FTP Route and eConnector (Process) components are identical expect that the Route component only stores the files and does not pass the files on, because the Route component is the last component in the AutoStore process. The Send to FTP eConnector (Process) component passes the files on to the next component in the process. You can also use the Send to FTP eConnector component to enable or disable document pass-through. Enabling this option makes the document available for other components in the process.

Features

The Send to FTP component offers the following features.

- Configuration of multiple FTP sites
- Active flow control
- Secured FTP site access
- Dynamic file renaming and storage

Using the Send to FTP component

The Send to FTP component is very similar to the FTP Store component but has some added functionality. The following are examples of how the Send to FTP component can be used.

- The Send to FTP component can be used in an enterprise where .PDF documents from several sources have to be available on an FTP server. These documents can then be processed for printing or to send as e-mail.
- Connect remote offices to a central office by using Send to FTP component. Scanners from all remote sites can easily be connected to a central site by using a secured FTP protocol.
- Use the bandwidth-control feature of the FTP component to control the bandwidth usage when remote scanning offices are connected to the central office. Bandwidth control lowers the negative impact that distributed scanning can create in a transmission infrastructure.

Configuring the Send to FTP component

Set the attributes of the FTP sites by using the options on the Send to FTP component.

- **Pass-through documents.** Select this check box if you want the documents to be available to the other components in your process.

NOTE This option is available only on the Send to FTP eConnector (Process) component, not on the Send to FTP Route component.

- **Add.** Click **Add** to add FTP sites to your process.
- **Server.** Specify the IP address or the server name of the FTP site. You can also specify the FTP site as `ftp://companysite.com`.
- **User Name.** Type the user name of the client who will have permissions to the FTP site. After you type the user name, type the password that is associated with this user. If you do not type the user name, the permissions are set to anonymous by default.
- **User Password.** Type the password that is associated with the user name.
- **Folder Path.** You can type the folder path where the documents will be stored on the FTP server. If you specify a new folder name in the dialog box, a new folder with the same name is created on the FTP server. However, if you do not specify a path or folder name, the data is stored on the root directory of the FTP server.
- **Overwrite Existing.** Select this check box if you want the newly created files on the FTP site to overwrite the existing files that have the same name. If you do not select this option, the file will automatically be appended with a number that is incremented each time a newer version of a file with the same name is stored on the FTP site.
- **Rename File.** Select this check box if you want the output file to be renamed.
- **Schema.** Type the Schema name for the output file name. You can use Runtime Replacement Tags (RRTs) to dynamically set the value of the schema.

Advance setting

Click the **Advanced** button in the **Configurations** dialog box to gain access to flow control, an easy method of controlling the speed at which the data is transmitted. The flow-control option manages the negative impact that FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. You can also test the flow-control speed and check the transmission speed.

NOTE To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits "dummy" data to the FTP site in order to test the transmission bandwidth.

The FTP server must be configured so that the account that is used to transfer files has write permissions to the FTP server. Some operating systems also require the account to have write permissions to the destination folder. For example, the if you are using the MS NTFS file system, then the account used to transfer files must be given write permissions by selecting the appropriate Security settings for that folder. The account must also have the appropriate account access permissions configured in the FTP configuration option in IIS Administrator.

Use the controls that appear to adjust the following parameters on the FTP:

- **Activate Flow Control.** Select this check box to control the speed at which the data is transmitted. The flow-control option manages the negative impact that FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. You can also test the flow-control speed and check the transmission speed.

  Select the Activate Flow Control check box to activate the flow-control options.

  - **Transfer Rate in Milliseconds.** Use this control to set your preferred transfer intervals in milliseconds. The longer the interval, the slower the transmission rate.
  - **Buffer Size in Bytes.** Use this control to set the size of each transmitted buffer. The larger the buffer size, the higher the transmission rate and the larger the impact on the bandwidth.
- **Run Test.** The test is conducted against your FTP site. Make sure that you have specified the correct FTP site, user name, and password. The tests are conducted using 100 Kb buffers, and at the end of each test cycle the results are reported in 100 Kb/X second, where X is the number of seconds it takes to send 100 Kb. Normal black-and-white documents are between 25 to 50 Kb. The size of images can vary depending on the type of scanner settings, and you should observe your scanner-setting parameters. Note that you must have write permissions to conduct FTP testing against a site.
- **Stop.** Use the **Stop** button to terminate the test-buffer transmission.

NOTE To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits "dummy" data to the FTP site to test transmission bandwidth.

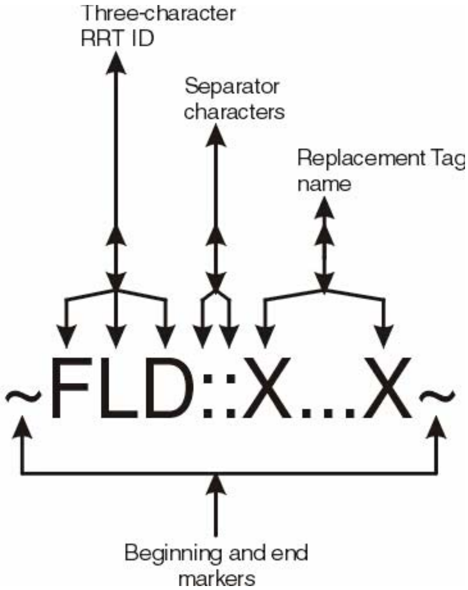
- **Remove.** Select the FTP site that you want to delete, and then click **Remove**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FTP**.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for the **Schema** field of this component.

Name	Description
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value that is concatenated with a name provides a unique file name.
FileExt	This is the original file extension value.
Path	This is the folder path associated with an FTP server. For example, if you have configured three FTP servers or folder paths, then this RRTN can take the following values: Path1, Path2, and Path3. Path1 refers to the first path entry configured, Path2 refers to the second path entry configured, and so on.

The following is an example of the RRTN process:

~FTP::FileName~~FTP::Counter~

The value Document5 is assigned if the original file name was "Document" and four (Document1 do Document4) files named "Document" were already within the destination folder path.

NOTE

The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~FTP::FileName~**, **~FTP::Counter~**, or **~FTP::FileExt~** with any other component except the Send to FTP eConnector (Process) component, and it must be used with the **Rename** field.

NOTE

You can create and display the counter with the required number of leading spaces and leading zeroes. For example, if the file name is TEST.DOC, and the rename schema is **~FTP::FileName~~%03FTP::Counter~~FTP::FileExt~**, then the resulting file names are TEST001.DOC, TEST002.DOC, and so on.

If the file name is TEST.DOC and the rename schema is **~FTP::FileName~~%3FTP::Counter~~FTP::FileExt~**, the resulting file names are TEST 1.DOC, TEST 2.DOC, and so on (note the two spaces after "TEST").

Field replacement tag name (FRTN). This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name

SSRTN	Description
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~FTP::%Y~-~FTP::%m~**" is replaced with "2004-10"

Troubleshooting

Problem	Solution
The files cannot be copied into the destination directory.	● Make sure that the destination FTP folder has write permissions. ● Make sure that the user name and password are valid. ● Make sure that the user has the appropriate permissions. ● Make sure that there are no firewalls. ● If **Overwrite Existing** is not selected, make sure that a file with the same name does not already exist.

Restrictions and limitations

This component has no known restrictions or limitations at this time.

Send to Folder

Use the Send to Folder component to copy files to any local or network directory. Using directories is one of the fastest methods of implementing a document-storage system. Storing files into flat folders requires no database and no software program, and results in lower costs. Use the Send to Folder component to create business rules for folder location, security access, and file naming of scanned images and processed files.

The Send to Folder component can also check your security access before storing documents into folders. When the check security feature is activated, you can store documents in a destination directory only if the administrator has assigned you an appropriate access level. By using this feature, an organization can create a secured storage location based on security settings.

The Send to Folder Route and eConnector (Process) components are identical except that the Send to Folder Route component only stores the files and does not pass the files on to other components because the Route component is the last component in the process. The Send to Folder eConnector (Process) component passes the files on to the next component in the process. You can also use the Send to Folder eConnector component to enable or disable document pass-through. Enabling this option makes the document available for other components in the process.

NOTE This component requires Active Directory Services with minimum of NT 4.0.

Features

You can use the Send to Folder component to accomplish the following tasks:

- Create dynamic folder names and locations.
- Rename the scanned files based on the document index information, field tabs, or runtime replacement tags (RRTs).
- Check the user security level against the destination folder location. Only users who have write permission can store files into a destination location.
- Specify additional information about documents that are stored in the folder by using keywords and comments.

  The Send to Folder component is fully integrated with other components such as barcode, form recognition, OCR, PDF, and so on. For example, a barcode value can be used as part of a destination folder name to create a dynamic destination folder based on the barcode values that are on the document. All types of files, including images, can be processed through this component.

Using the Send to Folder component

The following are two common scenarios for using the Send to Folder component:

- You log onto a device, scan a document, and place the document in your directory on a shared network drive.
- You store information in a folder where another process can read and use the information.

Configuring the Send to Folder component

Use the options described here to define the storage structure of documents.

- **Pass-through documents.** Select this check box if you want the documents to be available to other components in your process.

NOTE This option is available only on the Send to Folder eConnector (Process) component, not on the Send to Folder Route component.

- **Add.** Click **Add** to add a folder path to your process.
- **Remove.** Select the folder path that you want to delete, and then click **Remove**.

General tab

Use the options on this tab to set the following attributes.

- **Folder Path.** Use the text box to identify the destination-folder path. The AutoStore server must have write authorization to the folder that you identify. The Capture component can dynamically set the destination folder path.
- **Overwrite Existing File.** If you select this check box, the program overwrites files that have the same name. If you do not select this check box and a file that has the same name exists in the destination-folder path, an error message is generated.
- **Rename File.** Select this check box if you want to rename the output file.
- **Schema.** Use this text box to set the schema that is used to reformat the output file name. You can use runtime replacement tags (RRTs) to set the value of the schema dynamically.
- **Check User Security.** If you select this check box, the software checks the user's or the sender's security level to verify the write access. This refers to a lookup in the user's Active Directory Services to find out if the user has security rights to copy the files to the designated folder.
- **User Name.** Type the name of the user for the directory.

Summary tab

Use the options on this tab to set the document attributes. These options appear when you select the file and right-click **Properties**.

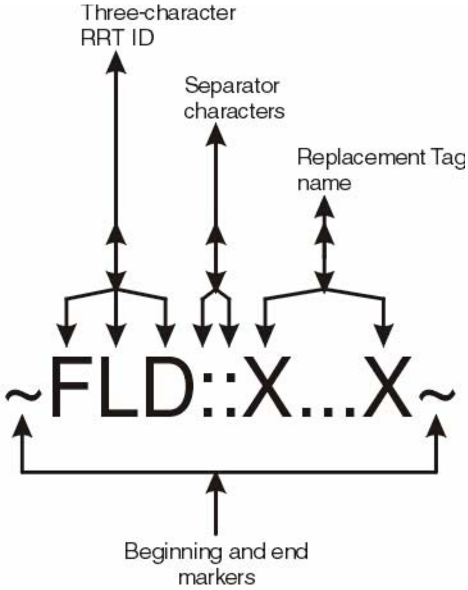
- **Subject.** Type a subject for your document.
- **Title.** Type the title of your document.
- **Author.** Specify the name of the author of your document.
- **Category.** Specify a category for your document.
- **Keywords.** Type keywords that are associated with your document. The keywords assist with future searches for the document. You can use a comma or a space to separate the keywords.
- **Comments.** Type comments that provide necessary information about your document.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	**Description**
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **STF**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for the **Schema** field of this component.

Name	**Description**
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value that is concatenated with a name provides a unique file name.
FileExt	This is the original file extension value.
Path	This is the folder path entry. For example, if you have configured three folder paths, then this RRTN can take the following values: Path1, Path2, and Path3. Path1 refers to the first path entry configured, Path2 refers to the second path entry configured, and so on.

The following is an example of the RRTN process:

~STF::FileName~~STF::Counter~

The value Document5 is assigned if the original file name was "Document" and four (Document1 do Document4) files named "Document" were already within the destination folder path.

NOTE The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~STF::FileName~**, **~STF::Counter~**, or **~STF::FileExt~** with any other component except the Send to Folder eConnector (Process) component, and it must be used with the **Rename** field.

NOTE You can create and display the counter with the required number of leading spaces and leading zeroes. For example, if the file name is TEST.DOC, and the rename schema is **~STF::FileName~~%03STF::Counter~~STF::FileExt~**, then the resulting file names are TEST001.DOC, TEST002.DOC, and so on.

If the file name is TEST.DOC and the rename schema is **~STF::FileName~~%3STF::Counter~~STF::FileExt~**, the resulting file names are TEST 1.DOC, TEST 2.DOC, and so on (note the two spaces after "TEST").

Field replacement tag name (FRTN). This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~STF::%Y~-~STF::%m~**" is replaced with "2004-10"

Troubleshooting tips

Problem	Solution
An error dialog box appears when you attempt to create a folder path.	Make sure that you have not used invalid characters in the folder path definition. Invalid characters are \, /, :, *, ", <, >, and |.
The Path RRTN is not replaced with a path folder.	This situation occurs when you specify a Path, such as Path8, where only six folder-path entries are configured with the Send to Folder component. Check to make sure that the Path number is valid.

Restrictions and limitations

This component has no known restrictions or limitations at this time.

ApplicationXtender (Process) component

Use the ApplicationXtender Process component to store documents into the Documentum ApplicationXtender, a content-management software package that brings large amounts of data online in a cost-effective manner.

The ApplicationXtender component provides robust and comprehensive security to protect sensitive business information. You can limit access to information within applications, which can further protect confidential information. In addition, user privilege security is provided so that users can be restricted from performing specific functions.

The ApplicationXtender component provides comprehensive electronic file management capabilities and supports a wide range of electronic content.

Feature highlights

Perform the following tasks by using the ApplicationXtender component features.

- Secure your user name and password.
- Select an application into which you want to store your documents.
- Set the title, subject, author, keywords, and comments that are associated with your documents.
- Place a document into a queue for further processing.
- Set the field values for the application that you selected.

The ApplicationXtender component is typically used with the Digital Sender or Poll Directory Capture components. Capture the files by using a Capture component, and then process them by using the ApplicationXtender component. This component can process any file type.

Using ApplicationXtender

This is an example of how to use the ApplicationXtender component:

The Digital Sender device captures content data once and routes it to the SharePoint Portal server. Create your process with the Digital Sender Capture component. Use the ApplicationXtender Process component to manipulate the captured data within the ApplicationXtender repository and save the data to the SharePoint Portal server. This AutoStore process provides a uniform capture capability across the enterprise and storage to the application or media that you select.

Licensing

Three types of licenses are available for this component:

- **Evaluation.** A 30-day fully functional component is available upon first installation.
- **Licensed.** The fully licensed component provides full capabilities indefinitely.
- **Expired.** After the evaluation period, unlicensed components expire without any further processing.

Configuring the ApplicationXtender component

Use the appropriate procedure to open the **ApplicationXtender** configuration dialog box to configure the ApplicationXtender component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the ApplicationXtender component.

Use the following options to configure the attributes for this component.

General tab

Before using the ApplicationXtender component to manage a document file, you must specify a valid DSN, user name, and password.

- **DSN.** Type the data source name to which you want to connect.
- **User Name.** Type the appropriate user name for the selected data source. If the ApplicationXtender component is using the Windows NT security provider, you must precede the user name with a domain name and a slash symbol. For example, documentation\rfrost indicates that "rfrost" is a user on the "documentation" domain.
- **Password.** Type the password that corresponds to the user name that you specified.
- **Application.** An application is the highest level of organization in the ApplicationXtender component. Use the application for storing and retrieving documents.

  Every time you store a document in an application, you must type index information for that particular document into the index fields. The ApplicationXtender component stores the index information in a database so that you can search it later to retrieve documents.
- **Pass-through.** This option is only available on the Application Xtender eConnector (Process) component. If the **Pass-through** option is activated, the documents are passed on to the next component in the process.

Attributes tab

If the computer on which the ApplicationXtender component is installed has been configured to allow searching by open document-management API (ODMA) attributes, you can search for documents by title, subject, author, keywords, comments, and the user name under which the document was created.

- **Title.** Type the title of the document.
- **Subject.** Type the subject of the document.
- **Author.** Type the name of the author of the document name.
- **Keywords.** Type the keywords for the document. Make sure that the keywords are separated by a comma.
- **Comments.** Type comments about the document.

Queue tab

You can place a document into a queue for further processing. The processing queues are used for batch OCR, full-text indexing, and printing.

- **Submit document to queue.** Select whether or not a document should be placed into a queue. If you decide to place the document into a queue, you need to provide a queue name.
- **Queue Name.** Select the name of the queue that you want to place the document into, after storing it into the ApplicationXtender component.
- **Description.** Type a description of the job in which the document will be submitted to a specified queue.

Field Values tab

When a document is added to an ApplicationXtender application, specify data for each of the index fields in the application. Each index field that is defined will be used to hold descriptive information about the documents that are stored in the application.

- **Field.** Type the field name.
- **Type.** Specify the field type. You can select from Text, Integer, Date, Boolean, and so on.
- **Required.** Specify whether or not the field is required. The process cannot be saved until the required fields are assigned a value.
- **Value.** Type the value of the field.

Using Knowledge Package Loader to configure the ApplicationXtender component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the ApplicationXtender component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the ApplicationXtender component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the ApplicationXtender component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the ApplicationXtender component

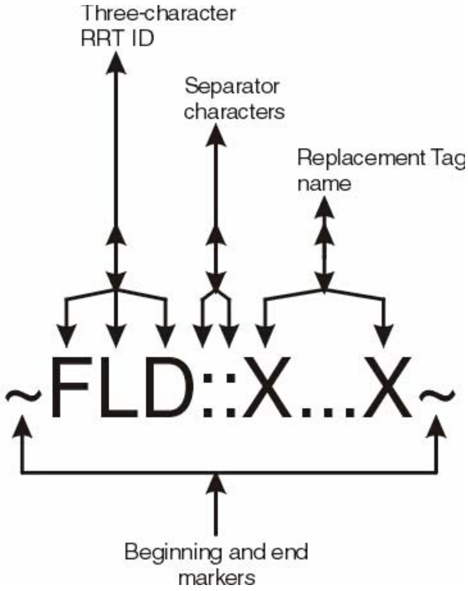
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the ApplicationXtender component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for the ApplicationXtender Process component is APX.

NOTE The ApplicationXtender Route component does not generate RRTs.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
ID	The identifier of the document stored in the ApplicationXtender component.

~APX::ID~ Replaces the identifier value referring to a particular document stored in the ApplicationXtender component.

For example, create an AutoStore process with an MFP Capture component, followed by the ApplicationXtender process component and the Send to Mail Recipient Route component. In the Send to Mail Recipient component, place **~APX::ID~** as the subject of the e-mail. This process sends e-mail to the designated recipient with the ID of the document just stored into the ApplicationXtender component as its subject. This e-mail and all other e-mails sent from this process can later be used as an inventory of sorts, of the documents successfully stored into ApplicationXtender. Later, if you want to reference the document, you can search the ApplicationXtender component for the document with this specific ID.

Troubleshooting tips

Problem description	Solution
No applications appear in the Application field drop-down list.	Make sure that the user name and password are valid. Make sure that you created an application that has corresponding user rights in the Application Generator.

Problem description	Solution
An error message appears indicating that you have to specify a value for all required fields.	Make sure that all of the fields that have the value YES under **Required** have a field value.

Restrictions and limitations

- This component is compatible with ApplicationXtender Version 4.x.

Notification component

Use the Notification component to alert users about the status of a job. Job status can be a success or failure. You can configure the Notification component to send e-mail messages, with or without attachments, about the job status. You can configure the e-mail notification to be sent on success or failure of a job.

The Notification component uses SMTP for sending e-mail. The SMTP server can reside locally on the computer where the AutoStore process is running. The SMTP server can also be located remotely, as long as the processing computer can communicate with the SMTP server.

Feature highlights

You can perform the following tasks with the Notification component.

- Send e-mail notification upon success or failure of a process.
- Include attachments in the notification e-mail message.
- Define multiple recipients of the notification e-mail.
- Define the subject and message of the e-mail notification.
- Use RRTs to capture job-related information within the message body.

Using the Notification component

The position of the Notification component in a process impacts the attachments that are sent in the notification e-mail. The attachment files originate from the component(s) that immediately precede the Notification component.

For example, if you configure the Notification component to send an attachment with e-mail, and you have an MFP source component and a FineReader OCR, the documents that are attached to the notification e-mail are the documents that are processed by the FineReader OCR and received from the MFP source component.

Configuring the Notification component

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Notification component.

Specify the SMTP server attributes and notification e-mail attributes in the **Notification Configuration** dialog box.

- **Notify on Success.** Select this check box to activate the e-mail notification when the process has been completed successfully. Then specify the e-mail attributes for sending the e-mail notification.
- **Notify on Failure.** Select this check box to activate the e-mail notification when the process fails (for example, the document was not saved, or FTP failed with the file). Next, specify the e-mail attributes for sending the e-mail notification.

- **Recipients.** Specify the SMTP e-mail addresses for individuals that are to be notified of the job status. You can specify multiple recipients in this field by using a semicolon as a separator.
- **Subject.** Specify the subject of the notification e-mail message.
- **Message.** Specify the body text of the e-mail message to be sent.
- **Attachment.** Select this check box if you want to attach document(s) to the e-mail notification. The attachment depends on the position of the Notification component in the process, because the attachment document(s) originate from the components that precede the Notification component. If you do not attach documents to the e-mail notification, the position of the Notification component is irrelevant.
- **SMTP Server.** Type the appropriate SMTP server network name or IP address. This SMTP server is used for the confirmation e-mail notification. SMTP default port 25 is used for sending e-mails to the server.
- **User name.** Specify the user name to log on to the SMTP server (if required).
- **Password.** Type the password to log on to the SMTP server (if required).

Use the appropriate procedure to open the **Notification Configuration** dialog box to configure the Notification component.

Using Knowledge Package Loader to configure the Notification component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Notification component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Notification component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Notification component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Notification component

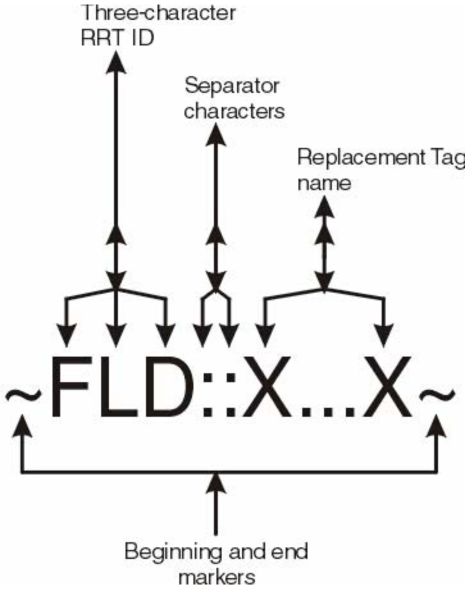
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Notification component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs; however, all of the parameters can contain RRT strings. For example, when MFP (4100/9000) is the Capture component, and the device Authentication option is turned on via the Secured Access or the Communications option, the **Recipients** parameter can be set to "**~M94::SenderAddress~**".

Troubleshooting tips

Problem description	Solution
Files are not attached to the e-mail notification.	Check the position of the Notification component in the process. Make sure that this component is placed where you want the attachment to be obtained from. The attachment originates from the component preceding the Notification component.
The e-mail message is taking too long to arrive.	Check to make sure that the SMTP server does not have a delay in processing the message. The SMTP server that is used to deliver the notification might not do so immediately, but might be configured to wait a specified amount of time before delivery.

Restrictions and limitations

- The Notification component is supported in Windows 2000 only.

Professional Barcode component

Use the Professional Barcode component to read barcodes from images. After a barcode is read, its information can be used by other downstream components within the configuration. Also, this information can be used as a deciding criterion for whether or not to split a document.

The Professional Barcode uses ABBYY Technologies for barcode detection.

Feature highlights

The values from the barcode can be used as parameters in subsequent components in the configuration. They can also be mapped into fields that will automatically be used by Capture or Process components that can use this field information.

The most common input types are black-and-white raster images that have been generated from a scanning device.

This component is commonly used to read barcodes that appear in black-and-white .TIF files that a scanning device has generated. The image must have at least a 200-dpi resolution. Depending on the scanner quality, and the size and type of the barcode, you might need to generate higher-quality images in order for the barcodes to be read correctly.

If you want to set up your process so that other components can use the barcode values, configure this component through a blocking component, such as Digital Sender, MFP, or POP3 E-mail. If you plan to use this component with a non-blocking Capture component, place a Data Interchange Process component before the Basic Image Management component.

Using the Professional Barcode component

This component is used as part of business processes that rely on barcode coversheets or barcode information inside a document (for example, when a barcode is used to represent the invoice number in an invoice document). You can also use this component to increase the speed when scanning large number of documents. By using the barcode separator sheets, the scanner operator can scan multiple documents because the separator sheets indicate the separation points to the scanner.

This component can be used to read the values of a barcode and then store the information together with the document in one of the supported document-management systems.

The information can also be used to dynamically set other values in the configuration. For example, if the invoice is being sent to a Folder Store component, the final file could be renamed by using the invoice number as part of the new name.

Configuring the Professional Barcode component

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Professional Barcode component.

The following attributes are available in the **Professional Barcode** configuration dialog box.

General tab

Use the options on this tab to specify the barcode attributes.

- **Activate.** Select this check box to turn on the barcode processing function.
- **Barcode Type.** If you know the barcode type that is going to be read, select it. If not, select autodetect and the component will read any of the following supported barcodes:
  - **Code 128.** A very high-density alphanumeric barcode that contains 106 different printed barcode patterns.
  - **Check Code 128.** The same as Code 128, but with a checksum character.
  - **Code 3 of 9 (Code 39).** An alphanumeric barcode that is simpler than Code 128.
  - **EAN 13.** European Article Numbering that consists of 13 numbers.
  - **EAN 8.** European Article Numbering that consists of 8 numbers.
  - **Interleaved 2 of 5.** A numeric-only barcode that is used for interleaving pairs of numbers in a high-density barcode format.
  - **Check Interleaved 2 of 5 (25).** Same as Interleaved 2 of 5 with a checksum character.
- **Barcode Orientation.** Select the barcode orientation by using the following options:
  - **Autodetect.** Detect the barcode orientation automatically. Use this when the image scan orientation can change, and the barcodes can appear upside down.

NOTE The following barcode orientation options allow faster processing, because no orientation detection is required and all barcodes are assumed to be uniformly aligned.

  - **Top to Bottom.**
  - **Bottom to Top.**
  - **Right to Left.**
  - **Left to Right.**

- **Search Zone.** Restrict the image search area manually by specifying the coordinates, either in inches or millimeters. Use the following syntax in the text box:
  - **Top.** The distance from the top of the page to the beginning of the search area.
  - **Left.** The distance from the left side of the page to the beginning of the search area.
  - **Right.** The distance from the left side of the page to the end of the search area.
  - **Bottom.** The distance from the bottom of the page to the end of the search area.

  You can specify the search zone by graphically selecting the zone from a sample image file. To do this, click "**…**" and select a sample image file from which to identify a search zone, and then click **Open**. You will be presented with a preview of the first page of the image. Move the mouse to the preview area. The cursor will change appearance. Click on the image and drag the cursor to select a square section of the image. Click **OK**. This is the area where the component will search for barcodes. You can change the file that you use for previewing by clicking **Open** and selecting a different image.
- **Page Split.** Select the page split attributes by using the following options:
  - **Split on Barcode Value.** Select this check box to split the document every time that the component detects the barcode that is specified in the edit box. The barcode value can contain wildcards such as ***** and **?**.

    For example, if "*" is specified in the edit box, then the document splits when it finds any barcode. If **1000?** is specified, then the document splits if the barcode is **10002**, **10003**, and so on.

    By definition, the following wildcard characters can be used:

    ***** - Any value of any length

    ? - Any single character
  - **Keep Barcode Page.** If the split is based on the barcode, you can decide to either keep the barcode page or discard it.
  - **Pre Separator.** Select this check box when you are using the barcode page as a separator and you want the barcode to serve as a separator at the beginning of the document. If this check box is not selected, the component assumes that the barcode separator is the last page of the split document and starts a new file after it.

Field Values tab

Use this tab to delete field entries from the list of field values.

- **Insert.** Click this button to type the field name and the barcode value that is associated with this field. Click "**...**" to provide the page number and barcode number of the barcode that contains the information that should be assigned to this field. You can also use RRT strings to dynamically specify the barcode value.
- **Delete.** Click this button to remove a field value entry from the field value list.

Using Knowledge Package Loader to configure the Professional Barcode component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.

3. In the **Component Name** window, select the Professional Barcode component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Professional Barcode component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Professional Barcode component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Professional Barcode component

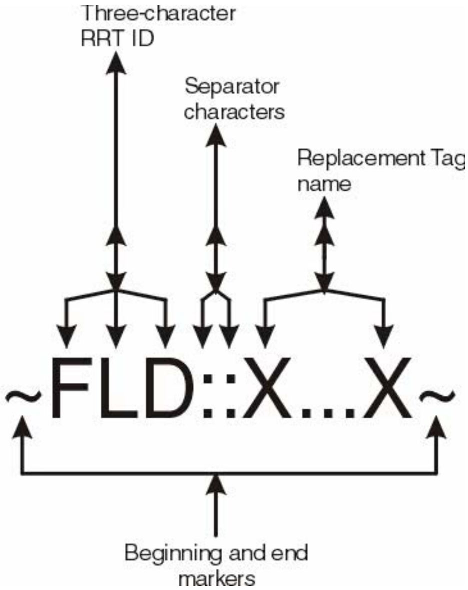
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Professional Barcode component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The values for single barcodes read in different pages can be used to dynamically configure other components in the configuration. By using different variations in referencing the page number and location where the barcode is, you can not only refer to a single value but also to the concatenation of several values or even all of the barcodes that are read for a document.

The following syntax is used to refer to barcode values that were read from the image file:

~FRB::PageNumber, BarcodePosition~

PageNumber is the image page where the barcode is located and **BarcodePosition** is the position of the barcode on the page (counting from left to right and from top to bottom).

Certain special characters can be used instead of a specific **PageNumber** or **BarcodePosition**:

*:: Any page number or barcode position.

-1: Last page or barcode position.

The possible combinations are shown in the following table.

~FRB::*, BarcodePosition~	This returns the concatenation of all of the barcode values in **BarcodePosition** found on any page of the image file.
~FRB::PageNumber,*~	This returns the concatenation of all of the barcode values for any position found on the **PageNumber** page of the image file.
~FRB::-1,-1~	This returns the barcode value of the last barcode on the last page of the image file.
~FRB::-1, BarcodePosition~	This returns the barcode value in **BarcodePosition** found on the last page of the image file.
~FRB::*,*~	This returns the concatenation of all of the barcode values for any position found on any page of the image file.

NOTE If you are splitting on the barcode, then the term "last page" refers to the last page of an image file after it has been split.

Troubleshooting tips

Problem description	Solution
The Professional Barcode component cannot read one of the supported barcode types.	The search zone does not include the area where the barcode is located. Using the image that contains the barcode as a sample, change the search zone so that it includes the location where the barcode appears. If you are not sure if this location is always going to be the same for all images, then do not use a search region. The component will search the entire page for the barcode.
The status monitor shows that the barcodes are being read but the barcode values references are not being replaced.	The service cannot detect the correct program that is responsible for handling the print operation for the specified file in the working folder. The Capture component is not a blocking component and you are not using a Data Interchange Process component in your configuration. Insert the Data Interchange component in your configuration before the Professional Barcode component, and configure the remaining components through the Data Interchange component.

Restrictions and limitations

- The splitting functionality is available only when reading multi-page .TIFF images.
- The following is a complete list of the file formats that the Professional Barcode component supports.

BMP

- 2-bit - uncompressed black and white
- 4- and 8-bit - uncompressed Palette
- 16-bit - uncompressed Mask
- 24-bit - uncompressed Palette and TrueColor
- 32-bit - uncompressed Mask

PCX, DCX

- 2-bit - black and white
- 4- and 8-bit - gray

JPEG

- Gray and TrueColor

TIFF

- Black and white - uncompressed, CCITT3, CCITT3FAX, CCITT4, Packbits
- Gray - uncompressed, Packbits, JPEG
- TrueColor - uncompressed, JPEG
- Palette - uncompressed, Packbits
- Multiimage TIFF

PNG

- Black and white, gray, color

Professional Image Management component

The Professional Image Management component is a Process component that is primarily used to enhance the quality and appearance of black-and-white images.

Use this component to enhance the quality of your scanned or faxed documents by performing operations that remove imperfections. You can also use the options in this component to split images into several documents based on a predefined set of criteria.

The Professional Image Management component uses LeadTools Technology for the different image-processing options.

If you have a process that receives an image file that contains several documents, each of which contains an equal number of pages, and these files need to be stored separately, you can use the Split function on a specific number of pages to implement batch processing and increase efficiency.

Feature highlights

The Professional Image Management component includes the following features:

- The Despeckle option removes specks from black-and-white images, such as fax transmissions or scanned documents.
- The Deskew option automatically straightens scanned images.
- The Smooth Text option smooths out the bumps and fills in the nicks in a black-and-white image.
- The Dot Removal option finds and removes dots and specks of various sizes. It is similar to the despeckle option but allows more control over which dots are removed. Certain images might have larger dots that would not be removed by the Despeckle option, but which can be removed by using this option.
- The Remove Hole Punches option removes hole-punch marks from a black-and-white image. (These marks usually result from scanning a document.) This will remove hole-punches marks that are located at the top, bottom, left, or right side of the image.
- The Line Removal option removes horizontal and vertical lines in a black-and-white image.
- The Border Removal option removes black borders from black-and-white images. (These marks are often the result of scanning documents where the paper document is smaller than the resulting scanned image.)
- The Remove Empty Pages option removes pages that do not contain any text or images. Note that "empty" does not mean a white page. However, a completely black page (or a page that consists of only any color other than white) is considered empty.
- The Split feature splits the original file into multiple files.

Configuring the Professional Image Management component

Use the appropriate procedure to open the **Image Management** configuration dialog box to configure the Professional Image Management component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Professional Image Management component.

The following attributes are available in the Image Management configuration dialog box.

Clean Image tab

Use the options on this tab to set the following attributes:

- **Activate.** Select this check box to activate the image management configuration.
- **Deskew.** Select this check box to deskew (straighten) the image.
- **Despeckle.** Select this check box to despeckle (remove small marks from) the image.
- **Smooth Text.** Select this check box to smooth the bumps and fill in the nicks in a black-and-white image.
- **Trim Edges.** Select this check box to trim blank space from the edges of scanned documents.
- **Remove Dots.** Select this check box to remove dots and speckles of various sizes. It is similar to the despeckle option but allows more control over which dots are removed. Certain images might have larger dots that would not be removed by the despeckle option, but which can be removed by using this option. Note that certain portions of an image that you want to retain might be removed if the maximum dot height and width are set too high.
- **Remove Hole Punches.** Select this check box to remove hole-punch marks from a black-and-white image. The hole-punch marks can be located at the top, bottom, left, or right side of the image. The default location is the left.
- **Remove Lines.** Select this check box to remove unwanted horizontal and vertical lines in a black-and-white image by specifying the line attributes. The maximum line width can be configured to be between 0 and 0.1 inch.
- **Remove Borders.** Select this check box to remove black borders from black-and-white images. The borders occur when scanned documents are smaller than the resulting scanned image.
  - **White Noise** specifies the amount of white noise tolerated when determining the border.
  - **Variance** specifies the amount of variance that is tolerated in the border.
  - **Border Area** specifies the percentage of the page from each edge that is designated as the search area for borders to remove.
- **Remove Empty Pages.** Select this check box to remove pages that do not have any text or images. Note that empty does not mean a white page. A completely black page (or any other color) will be considered empty (blank) as well. This option can be used in multipage .TIF files. It does not apply to any of the other supported formats by the different options of this component.

Split tab

This tab provides the following options for splitting an image.

- **None.**
- **Split into specified pages.** Specify the number of pages into which the document is split. For example, if you are scanning batches of three-page invoices, the process could receive one file containing several invoices. You can choose to split every three pages, resulting in a single file for each invoice.
- **Split on pages.** You can provide specific page numbers where the document should be split. You can instruct the component to split the document on a particular page or on several pages in a comma-separated list. If a page number does not exist, the component ignores it.
- **Split on empty page.**

NOTE

The **Split** option is used for splitting multi-page .TIFF files. It does not apply to any of the other supported formats by the different options of this component.

Using Knowledge Package Loader to configure the Professional Image Management component

1. Double-click the Knowledge Package Loader Capture component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Professional Image Management component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Professional Image Management component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Professional Image Management component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Professional Image Management component

1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).

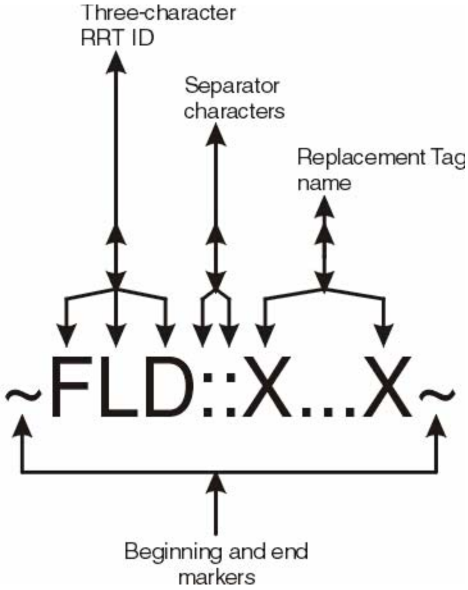
4. In the **Component Name** window, select the Professional Image Management component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.

Segment name	Description
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

Use RRT strings when your attribute setting requires dynamic values from the data that is being processed. To set a component attribute to a value that an RRT string can reference, type the appropriate RRT string in the attribute. The AutoStore server replaces the RRT string with the referenced value from the data stream within each processed item.

This component does not have any RRT strings associated with it.

Troubleshooting tips

Problem description	Solution
When using the Line Removal option, some of the lines are not removed.	When some of the lines in the image are wider than the maximum width that is specified in the configuration, those lines are not removed. Increase the maximum specified width in the component configuration.

Restrictions and limitations

The following options can only be used in black-and-white images (1-bit formats):

- Smooth Text.
- Line Removal.
- Hole Punch Removal.
- Dot Removal.
- Border Removal.

The following are the supported formats for the Professional Image Management component:

JPEG Formats

- JPEG File Interchange Format.
- Tagged Image File with JPEG compression.
- JPEG 2000 Format. This file format contains image data and extra information about the contents and organization of the file.

GIF Formats

- CompuServe GIF.

TIFF Formats

- Tagged Image File Format, with no compression and with RGB color space and 8-bit grayscale.
- Tagged Image File, with no compression and with CMYK color space.
- Tagged Image File, with no compression and with YCbCr color space.
- Tagged Image File with PackBits Compression and RGB color space.
- Tagged Image File with PackBits Compression and CMYK color space.
- Tagged Image File with PackBits Compression and color YCbCr space.
- Tagged Image File with CMP Compression.
- Tagged Image File with JBIG Compression.
- Tagged Image File with a vector image saved as a DXF.
- Tagged Image File with JPEG 2000 Compression. This file format contains only a stream of image data.
- Tagged Image File with Wavelet CMP Compression.

BMP Formats

- Windows BMP, with no compression.
- Windows BMP, with RLE compression.
- OS/2 BMP version 1.*x*.
- OS/2 BMP version 2.*x*.
- Wireless Bitmap file. Type 0.

WMF and EMF Formats

- Windows Meta File.
- Windows Enhanced Meta File.

Exif Formats

- Exif file containing a TIFF image, with no compression and with RGB color space.
- Exif file containing a TIFF image, with no compression and with YCbCr color space.
- Exif file containing a JPEG compressed image.

1-Bit FAX formats

- TIFF, compressed using CCITT.
- TIFF, compressed using CCITT, group 3, 1 dimension.
- TIFF, compressed using CCITT, group 3, 2 dimensions.
- TIFF, compressed using CCITT, group 4.
- Raw FAX, compressed using CCITT group 3, 1 dimension.
- Raw FAX, compressed using CCITT group 3, 2 dimensions.
- Raw FAX, compressed using CCITT group 4.
- IOCA, compressed using CCITT group 3, 1 dimension.
- IOCA, compressed using CCITT group 3, 2 dimensions.
- IOCA, compressed using CCITT group 4.
- IOCA, compressed using IBM MMR, with the MO:DCA wrapper.
- IOCA, uncompressed, with the MO:DCA wrapper.

Other 1-Bit formats

- MacPaint.
- Portable Bitmap - ASCII File.
- Portable Bitmap - Binary File.
- XBitmap File.
- Microsoft Paint.

OpenText Livelink eConnector component

Use the OpenText Livelink component to capture documents and data into the OpenText Livelink 9.x application. The integration between AutoStore and Livelink enables you to capture documents from a variety of AutoStore Capture components into this document management system.

Log on to the OpenText Livelink component and identify the folder, category, and fields within the component where you want the document to be stored. Set up your AutoStore workflow process to capture content from the specified Capture component, and then store the content in the specified OpenText Livelink folder.

The integration between the AutoStore software and the OpenText Livelink component takes advantage of Livelink features, such as multiple category support, security options, multi-value document and folder naming support.

Features

You can perform the following tasks with the OpenText Livelink component.

- Support various document version control.
- Lock documents for enhanced security.
- Integrate the AutoStore software with the OpenText Livelink component to use the security options and features.
- Support multiple category and folder structure.

Using the OpenText Livelink component

Use the OpenText Livelink component to store any type of content from various sources. This component allows you to easily meet your regulatory and archival requirements for records retention. The following are examples of how the OpenText Livelink component can be used.

Knowledge Management. Capture, process, organize, share, and store valuable information into the OpenText Livelink 9.x document management system.

Direct device connectivity. Allow users to archive important documents by pressing a few buttons on a scanning device. Use the OpenText Livelink component to directly connect various devices such as digital copiers, desktop scanners, production high-speed scanners, desktop files, and other types of files to backend OpenText Livelink 9.x.

Batch Import Server. Use the OpenText Livelink component along with the Poll Directory component to create batch import directories where files read in from various directories can be imported directly into backend OpenText Livelink 9.x.

Uniform Capture Process tools. Create capture business rules that determine how your content is captured into a backend document management system by using the AutoStore process designer tools.

Connect e-mail files to the Livelink document management system. Use POP3 email or SMTP Capture components to connect e-mail content and archive all e-mails within an inbox or all e-mails sent to an SMTP gateway into the OpenText Livelink component for archival, management, or sharing.

Configuring the OpenText Livelink component

The following attributes are available in the **OpenText Livelink** configuration dialog box.

- **Preferences tab.** Configure the settings for the AutoStore process home directory, IP port number, and other administrative directories.

General tab

Use the attributes in this tab to define the connectivity to the OpenText Livelink application.

- **Server.**Type the IP address or the host name of the OpenText Livelink 9.x server.
- **Database.** Type the name of the OpenText Livelink database to which you are connecting. If you specify a null string, the system uses the default database that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
- **Username.** Type a valid OpenText Livelink user name. The processed documents are associated with this user name.
- **Password.** Type a valid password that corresponds with the user name.
- **Port.** Type the port number that the server uses to communicate with the OpenText Livelink server. The port number value must match the port number that is configured on the OpenText Livelink software.
- **Impersonate.** Type the user name of the user that you want to impersonate. You must use the administrator's user name and password.
- **Advanced.** Select this check box to activate the security mechanism that is used by the OpenText Livelink application programmer's interface (LAPI) software to exchange data with the OpenText Livelink server.

The following options are available in the **Advanced** attribute.

Direct Connection. This is the default option. A direct connection does not ensure the confidentiality of the data that is passed over the Internet or any unsecured network. The data messages are passed as unencrypted plain text. If another user intercepts the plain text message, that user can view the content.

Non-secure Tunneling. Non-secure tunneling occurs when a LAPI application exchanges data with an OpenText Livelink server by transmitting unencrypted (plain text) HTTP messages through the Web server that is integrated with an OpenText Livelink server.

- AutoStore sends data in an HTTP request to the Web server that is integrated with the OpenText Livelink server.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the OpenText Livelink server (similar to the functionality when an OpenText Livelink request is made over a socket connection).

- The OpenText Livelink server processes the AutoStore request, generates a response, and then returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which returns the response to AutoStore.

Field name	Description
HTTPUserName	This is the user name that is recognized by the Web server.
HTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	This is the URL to the OpenText Livelink CGI integration process.

When you have selected the **Non-secure Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	Default value	Description of what to use
Server	WebServerHost	This is the name of the computer on which the Web server is running.
Port	80	This is the non-secure port to the Web server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Non-secure Proxy Server Tunneling. Non-secure proxy server tunneling occurs when a LAPI application exchanges data with an OpenText Livelink server by transmitting unencrypted (plain text) HTTP messages through the Web proxy server, which passes messages to the Web server that is integrated with an OpenText Livelink server.

- AutoStore sends data in an HTTP request to the Web proxy server, similar to a Web browser requesting data from a Web server through the Web proxy server.

- The Web proxy server forwards the request to the Web server that is integrated with the OpenText Livelink server.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the OpenText Livelink server, (similar to the functionality when an OpenText Livelink request is made over a socket connection).

- The OpenText Livelink server processes the AutoStore request, generates a response, and then returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which returns the response to AutoStore through the Web proxy server.

Field name	value	Description
HTTPUserName	myHTTPUserName	This is the user that is recognized by the Web server.
HTTPPassword	myHTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	http://host:port/livelink/livelink.exe	This is the entire URL to the OpenText Livelink CGI integration process where the host is the OpenText Livelink host name and the port is the Web server port number.

When you have selected the **Non-secure Proxy Server Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	Default value	Description of what to use
Server	ProxyServerHost	This is the name of the computer on which the Web proxy server is running.
Port	8080	This is the port to the Web proxy server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Secure Tunneling. This option requires Livelink Secure Connect. You must purchase this product separately, and install Livelink Secure Connect before you can implement secure communications. Livelink Secure Connect includes RSA BSAFE cryptographic and security protocol software from RSA Security, Inc. The data is almost impossible to decipher if it is intercepted, but it is easily converted to plain text by the application that is intended to receive the data.

- AutoStore sends data in a Secure HTTP (HTTPS) request to the Web server that is integrated with the OpenText Livelink server. Browsers use the HTTPS protocol to encrypt user page requests and to decrypt pages that are returned by a Web server.

- The Web server negotiates the SSL connection with AutoStore via an SSL "handshake," and decrypts the application request.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the Livelink server (similar to the functionality of a typical OpenText Livelink request).

- The OpenText Livelink server process the AutoStore request, generates a response, and returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which encrypts the response and returns it to AutoStore.

Field name	**value**	**Description**
HTTPUserName	myHTTPUserName	This is the user that is recognized by the Web server.
HTTPPassword	myHTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	/livelink/livelink.exe	This is the entire URL to the OpenText Livelink CGI integration process.
CA root certificates		This is a secure LAPI client application that requires the root certificate of the Certificate Authority (CA) from the secure Web server that is integrated with an OpenText Livelink Server to verify the authenticity of the certificate that is passed. In most cases, third-party CAs provide instructions on how to obtain their root certificates on their Web sites (for example, www.verisign.com or www.entrust.com). You can also export some CA root certificates from Microsoft Internet Explorer 5.0 and later.

When you have selected the **Secure Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	**Default value**	**Description of what to use**
Server	WebServerHost	This is the name of the computer on which the Web server is running.
Port	443	This is the secure port to the Web server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.

Field name	Default value	Description of what to use
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Document tab

Use this tab to set the document attributes for the document that is stored in the OpenText Livelink application.

- **Description.** Type a description of the document that you are storing.
- **Folder.** Specify the OpenText Livelink folder destination where you want to store your document. You can dynamically create folders from this location.
- **Lock file.** Select this check box to limit availability of the document. When this check box is selected, only the user who is logged in can gain access to the document. If this check box is not selected, the document is available to all users.
- **Pass-through.** This option is only available on the OpenText Livelink eConnector (Process) component. Select this option when you want the document passed through to the next component in the AutoStore workflow process.
- **Replace.** Select this check box if you want the newly-processed document to replace the existing document.
- **New Version.** Select this check box if you want to store the newly-processed document without overwriting the existing document.
- **Rename.** Type the name of the newly-processed document. See the *Component RRT ID* section for more information.

System tab

Use this tab to gain access to additional node attributes that are configured on the OpenText Livelink server. These node attributes allow you to assign values to the attributes, which are applied to every document in the OpenText Livelink application.

Follow these steps to create new attibutes for the OpenText Livelink application:

1. Click **Administer Additional Node Attributes** in the **System Administration** section on the **Livelink Administration** page.
2. Click **Add a New Attribute Link** on the **Administer Additional Node Attributes** page.

3. Type a unique name for the attribute in the Name field on the **Add New Attribute** page.

4. Select one of the following attribute types from the **Type** drop-down list:

   - **Text: Field**
   - **Text: Popup**
   - **Text: Multiline**
   - **Flag: Checkbox**
   - **Date: Field**
   - **Date: Popup**
   - **Number: Text field**
   - **Number: Popup**

Categories tab

Use this tab to create and remove association between the processed document and the categories that are defined in the OpenText Livelink application.

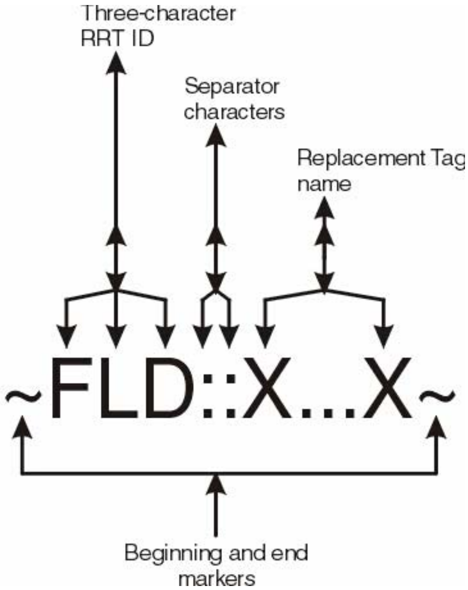
- **Select.** Click **Select** to view a list of all of the categories that are defined in the Enterprise workspace. You can associate your document with multiple categories. Select a category that you want to associate your document with and populate the field values if appropriate.
- **Remove.** Click **Remove** to remove the association between a document and a category.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **OTX**.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	This is the original file extension value.

The following is an example of an RRTN:

~OTX::FileName~~OTX::Counter is replaced by the value "Document5" if the original file name was "Document" and up to four (Document1 to Document4) files already existed within the destination folder path.

Field replacement tag name (FRTN). This component does not support FRTNs and replacement of field names with metadata values.

Special set replacement tag name (SSRTN). This component does not support SSRTNs.

Restrictions and limitations

- You cannot create categories through the OpenText Livelink component. A category must be created via the Livelink application configuration first.
- You cannot create fields through the OpenText Livelink component. Fields must be created via the Livelink application configuration first.

Troubleshooting

If no categories are available, either your user name and password are incorrect or you have not created any categories on the Livelink Enterprise Server.

Send to Database Process component

Use this component in a process to directly write images, files, or data into open database connectivity (ODBC) database tables such as Microsoft Access, or Visual Fox Pro.

Use this component to directly update structured query language (SQL) tables with images and data elements, or to create an archival system.

NOTE The ODBC data source must be set up correctly before you configure the Send to Database component.

Feature highlights

Use the Send to Database component to perform the following tasks:

- Select from various data sources where you can store data and binary documents.
- Provide the user name and password for access security.
- Select the table name (location) where you want to place the image and field values.
- Select the binary image field name (table column) into which you want to save an image.
- Add, modify, and remove field values to and from a database table.

Using the Send to Database Process component

The most common use of the Send to Database component is for archival purposes. You can store files, data, and documents in binary format in ODBC-compliant tables for long-term archival and record keeping. For example, you can use the MFP (4100/9000) as the Capture component in the APD, then use that component to configure the Send-to Database component. Assign RRTs to store the images, files, and new field names into the data source that you select.

The Send to Database component can be used with any file or document type.

Configuring the Send to Database Process component

Use the appropriate procedure to open the **Send to Database** configuration dialog box to configure the Send to Database component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Send to Database component.

The following attributes are available in the **Send to Database** configuration dialog box.

- **General** tab. Enter the following data source attributes on this tab.
  - **Data Source.** Click "**…**" for a list of available data sources. Click to select a data source. This is a required field.
  - **User Name.** (optional) Type the user name to be used for logging in to the Data Source.
  - **Password.** (optional) Type the password to be used for logging in to the Data Source.
  - **Table Name.** Type the table name to which you want to connect. This is a required field.
  - **Image Field Name.** Select the field name (table column) in which you want to save an image. **Image Field Name** has to be a binary large object (BLOB) field or the process will fail when it is run. This field must be configured.

  If you want to create new fields in the database, proceed to the **Field Values** tab and select **Add**.
- **Field Values** tab. Use this tab to create, modify, or remove fields in the database.
  - **Add.** Click this button to add new field value entries to the database table. The new field value can contain Runtime Replacement Tags (RRTs).
  - **Modify.** Click this button to modify the field value entries for the database table.
  - **Remove.** Click this button to remove a field value entries from the database table.

Using Knowledge Package Loader to configure the Send to Database (Process) component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Database component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Database (Process) component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Database component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Database (Process) component

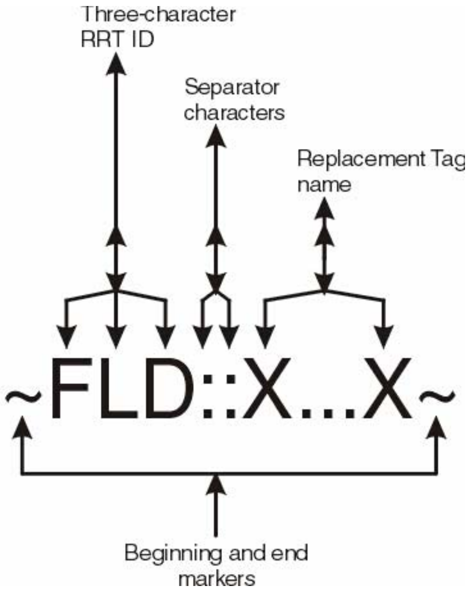
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to Database component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is SDB.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	The original file name value.
FileExt	The original file extension value.
FileSize	The size of the file in bytes.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)

SSRTN	Description
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~SDB::%m~-~SDB::%d~~SDB::%Y~**" is replaced with "08-20-2004"

Troubleshooting tips

Problem	Solution
An error appears when you click the **Table** browse button.	You either do not have a valid user name and password, or you have an invalid data source. Make sure that the data source that you selected is actually connected and is configured properly.
The "Duplicate output destination" error appears.	This error occurs when your process has the ABBYY FormReader v6.0 component along with the Send to Database component in the following scenario: ● You add a field value entry in the Send to Database component. ● The **Export All Fields** box in ABBYY FormReader v6.0 component is selected. ● An identical field name (to the one specified in the Send to Database component) is exported to the database. To resolve this error, change the field value entry in the **Field Value** tab of the Send to Database component.

Restrictions and limitations

- You cannot create tables dynamically from this component. If you want to create a database table, create it within the database.
- The data source, table name, and the image-field name fields are mandatory and must be configured.
- The image-field name must be a binary large object (BLOB) field.
- All images within a BLOB field are stored as binary values. You must develop your own image-extraction tool in order to extract and view the images.

Data Interchange component

Use the Data Interchange component to enable data to be interchanged between components when no mapping components are present within the AutoStore process.

When an AutoStore process does not include any mapping components, the process cannot share information, such as Runtime Replacement Tag (RRT) values. To enable the AutoStore process to use data interchange, add the Data Interchange component early in the process.

The Data Interchange component cannot be used with other mapping components within the same process. For example, if the AutoStore process uses the Data Interchange component and the MFP Component, the Data Interchange component will not provide any additional capabilities because the MFP component is a mapping component.

Feature highlights

The Data Interchange component makes it possible to interchange data between the other components that make up the AutoStore process.

Using the Data Interchange component

Use the Data Interchange component in an AutoStore process that would poll a directory of images with a barcode on the first page, read the barcode, and use the barcode value as the file name for the Folder Store component. A common mistake is to set up a process using the Poll Directory Capture component, the Professional Barcode Process component, the Folder Store Route component, and **~L1B::<1,1>~** as the RRT within the file name in the Folder Store component. The reason this does not work is because this process does not contain a mapping component and lacks the enabler to interchange data via RRTs.

The Data Interchange component acts as the mapping enabler component for any Capture component that is not natively designed as a mapping component and does not allow other component parameters to be mapped together at configuration within the APD.

When you introduce the Data Interchange component to the beginning of the process previously described, the process configuration mapping is enabled. Component configuration is performed through the Data Interchange component, which provides the ability to map configuration across the components and use RRTs to pass data in between each component. The modified process uses the Poll Directory component, followed by the Data Interchange component, the Professional Barcode component, and then the Folder Store component.

The Data Interchange component sets up the common memory configuration area that is accessible by all of the components, which enables the mapping and interchange of the RRTs. Without the Data Interchange component, none of the RRTs would be replaced with actual values.

Configuring the Data Interchange component

Use the appropriate procedure to open the **Data Interchange** dialog box.

Using Knowledge Package Loader to configure the Data Interchange component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Data Interchange component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Data Interchange component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Data Interchange component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Data Interchange component

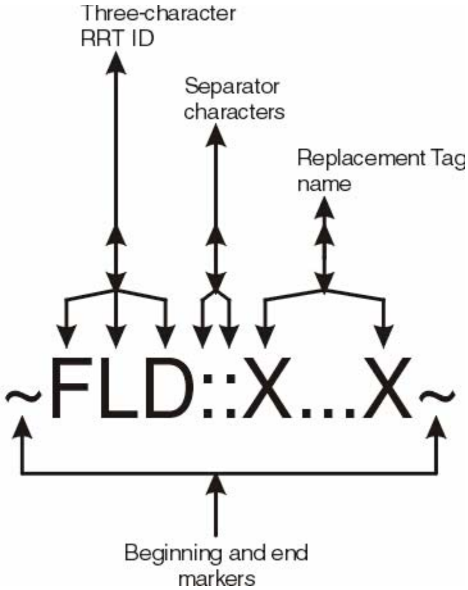
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Data Interchange component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs.

Troubleshooting tips

No information is currently available for this section.

Restrictions and limitations

This component has no known restrictions or limitations.

Hummingbird (5.x) Process component

Use the Hummingbird Process component to store documents into a document-management system where records can be identified and searched for by using database queries. The only difference between the Hummingbird Process and Route components is that the Pass-through field option on the **General** tab is only available with the eConnector (Process) component.

NOTE The Hummingbird component runs on a client computer that has access to the Hummingbird document-management server.

Feature highlights

You can perform the following tasks by using the features that the Hummingbird component provides.

- Select the library that you want to use.
- Select the profile that you want to use.

This component accepts any document or image file as an input type.

NOTE You must type a valid user name and password in order to populate the **Profile** field.

Using the Hummingbird component

Use the Hummingbird component to store any type of content from a variety of sources. You can meet your regulatory and archival requirements by using the Hummingbird component. The following examples show some common uses of the Hummingbird component.

- **Corporate content repository.** Use the Hummingbird component to protect valuable corporate information. If you want to store documents onto a server where other users (all of whom have permission rights to the files) can view the document, save these documents onto the server, and then give the other users permission to open and use the files.
- **Direct device connectivity.** Use the Hummingbird component with a scanning device to archive important documents. You can directly connect the Hummingbird component to devices such as digital copiers, desktop scanners, production high-speed scanners, desktop files, and other types of files, and then to the back-end Hummingbird document-management system.
- **Batch import server.** Use the Hummingbird component with the Poll Directory component to create batch import directories, where files that have been read from various directories can be imported directly into the back-end Hummingbird document-management.

- **Uniform capture process tools.** You can use the Hummingbird component process-design tools to create business rules that dictate how your corporate content is captured into back-end document-management systems.
- **Connect e-mail files to the back-end Hummingbird document-management system.** Use the POP3 E-mail component or the SMTP Capture component to connect e-mail content and to archive all of the e-mails within an inbox, or all of the e-mails that were sent to an SMTP gateway, in your back-end Hummingbird document-management system.

Licensing

Three types of licenses are available for this component: Evaluation, Licensed, and Expired.

- **Evaluation.** A 30-day fully functional component is available at the first installation.
- **Licensed.** The fully licensed component provides full capabilities.

NOTE Becoming fully licensed requires a Hummingbird client, which is not included.

- **Expired.** After the evaluation period, unlicensed components expire and will not process documents.

Configuring the Hummingbird component

Use the appropriate procedure to open the **Hummingbird** configuration dialog box to configure the Hummingbird component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Hummingbird component.

The following attributes are available in the **Hummingbird** configuration dialog box.

General tab

Use the options on this tab to set the general user attributes of the document-management server.

- **User Name.** Type the user name for connecting to the Hummingbird document-management server. This user must be able to gain access to various libraries and deposit documents. You can use RRT values to create a dynamic login user name based on the user who will be sending documents.
- **Password.** Type the password for the user name that you defined.
- **Domain.** Type the Windows domain name.
- **Logon Type.** Select the appropriate logon type, which is specified by the document-management server. Select from the following logon types that are available in the drop-down list: Library, Microsoft Network, Network Bindery, or Network NDS.

- **Library.** Select the library to which the user will connect. Note that the user must have the appropriate security permissions.

  When you select a library, two other text boxes become active, depending on which library you select. For example, if your library is the default library, then the text boxes remain inactive. However, if you select the Legal, Finance, or Government library, then the text boxes become active, so that you can set the other required features for those specific libraries.

  The Finance library requires **Account** and **Department** fields. The Legal library requires **Client** and **Matter** fields. The Government library requires **Organization** and **Department** fields.
- **Profile.** Select the profile that you want to use to store the document.
- **Typist.** Type the name of another user in the Hummingbird library. The value of this field *must* be another user in the Hummingbird library. The **Last Edited By** field in the document is set to this value.
- **Impersonate.** Type the name of the user that you want to impersonate. The user name that you specify here becomes the author of the document, instead of the user name that is specified in the **User Name** field. If you leave this field blank, the author is the user name that is specified in the **User Name** field. When you specify a user name in the Impersonate field, that author also receives security permissions.
- **Rename.** Select this check box and specify the schema that is to be used. If you do not select this check box, and the library does not take duplicate names, the document will fail to be stored if another document that has the same name already exists in that library.
- **Pass-through.** Enable this option to pass the document to the next component in the process.

NOTE

The **Pass-through** field is only available with the Hummingbird eConnector (Process) component.

Document tab

Use the options on this tab to set the following attributes.

- **Secure Document.** Select this check box so that only the user who is logged in and the designated **Typist** can see or make changes to the document. Otherwise, any user can gain access to the document.
- **Folder.** Select a folder to which you want to add the document. If you leave this field blank, the document is added to the root directory.
- **Add.** After specifying file extensions, click the **Add** button to add the extension and relate it to a file type. You can specify many file extensions and types.
- **Modify.** Select this button to change an existing file extension.
- **Remove.** Select an existing file extension and click the **Remove** button to delete the file extension.

Fields tab

The user creates the attributes listed here in the DM designer.

- **Fields.** Type the name of the field.
- **ID.** Specify the supported object type. Select from the following types: Edit, ComboBox, CheckBox, MediumEdit, RadioGroup, MultiEdit, and WideEdit.
- **Type.** Specify a supported type field. These types are supported: String, Date, Time, and Integer.
- **Required.** Type `Yes` if this is a required field, or `No` if this is not a required field.
- **Value.** Type the field value.

Using Knowledge Package Loader to configure the Hummingbird component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Hummingbird component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Hummingbird component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Hummingbird component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Hummingbird component

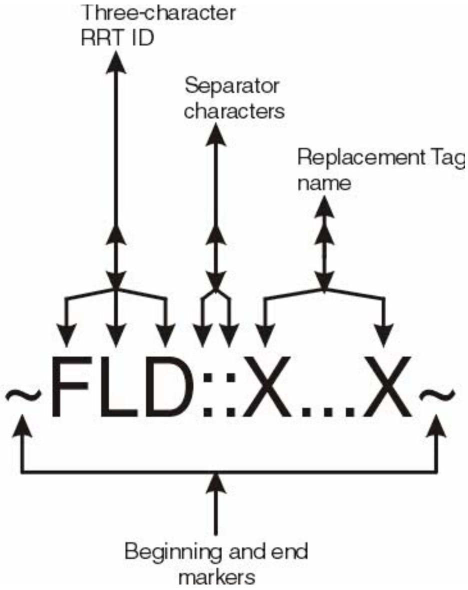
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Hummingbird component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **HUM**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter based on the duplicate file names found within a directory. The counter value concatenated with a name provides a unique file name.
ID	The ID number of the document that is stored in the Hummingbird component.
Library	The value of the library.
Domain	The domain name that is used when logging in to the Hummingbird component.

The following is an example of an RRTN:

~HUM::FileName~~HUM::Counter~

Replaced with the value "Document5" if the original file name was "Document" and up to four (Document1 to Document4) files already existed within the destination folder path.

NOTE

The RRTN values FileName and Counter can only be used with the **Rename** field of this component. You cannot use **~HUM::FileName~** or **~HUM::Counter~** in any other components except Hummingbird, and the RRTN must be used with the **Rename** field.

This rule *does not* apply to the RRTN values ID, Library, and Domain.

Field Replacement Tag Names (FRTN)

This component does not have any FRTNs.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem	Solution
The **Document Type** box is not populated.	Make sure that you have provided a valid user name, password, and domain, and the appropriate logon type and library.

Restrictions and limitations

- You might have to add one library at a time.
- You cannot select multiple libraries.
- If you store a file into the DM server without specifying a corresponding extension in the file extension list, the file will not be saved correctly.

Watermark component

Document protection is one of the key areas of any document archival and management system. By using the Watermark component, captured documents can be permanently watermarked with images, such as company logos or text, such as Confidential, Do Not Copy, and other appropriate words. When documents (files) are watermarked, the security information travels with the document as a permanent part of the document itself.

Watermarks can be created by using important indexing information such as case numbers (in litigation cases, for example) and other document- or page-specific information. When applied to the files, such watermark information can help maintain the index information as part of the document itself. You can also use VB/Java Scripts to look up watermark text and create a dynamic content-based watermark for your documents.

Feature highlights

Use the Watermark component to perform the following tasks:

- Apply permanent watermarks to image files.
- Apply permanent static or dynamic text.
- Apply a tile or stretch watermark to a portion of an image or to an entire image.

You can also use RRT tags from other components to create dynamic watermark values that are linked to the images and documents that are being processed. Dynamic watermarks provide a powerful method for customizing your watermarks based on content.

Using the Watermark component

Watermarks can be used in many ways within a business process. The following are examples of common usage scenarios.

- **Confidentiality marks.** Use the Watermark component to accurately reflect the confidential status of documents.
- **Status marks.** Use the Watermark component before you print a document to show its status (certified, non-certified, confidential, and so on).
- **Copyright.** Use the Watermark component to reflect copyrights and ownership of information.
- **Ownership marks.** Prevent unauthorized representation of information by using the Watermark component on the documents that you own.
- **Dynamic marks.** Based on the Capture content, select an RRT for the watermark value and create a dynamic watermark that ties the document to the business process. Examples of this use include watermarking documents with a user name, scanner ID, or other similar process-based variables.

NOTE Only raster image files being processed and graphical watermarks are supported.

Licensing the Watermark component

Three types of licenses are available for this component: Evaluation, Licensed, and Expired.

- **Evaluation.** A 30-day fully functional component is available upon first installation.
- **Licensed.** The fully licensed component provides full capabilities indefinitely.
- **Expired.** After the evaluation period, unlicensed components expire without any further processing.

Configuring the Watermark component

Use the appropriate procedure to open the **Watermark** configuration dialog box to configure the Watermark component.

Set up the Watermark component attributes to achieve your business-process requirements, and then add a Route component. The image files are automatically converted to the required output format and delivered to the Route component.

Use static or dynamic values as defined in the Source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Watermark component.

The following attributes are available in the **Watermark** dialog box:

- **No Watermark.** Select this check box if you do not want watermarking for a specific Route component. Use this check box to activate or deactivate this component within a process. For example, if you are using an MFP device that has multiple configuration routes, you might enable or disable watermarking for specific routes.
- **Image watermark.** Select this check box if you want to use an image such as a company logo as the watermark. If this option is selected, use the **Select File** button to select the image file.
- **Text watermark.** Select this check box if you want to use text such as "Confidential" or "Do Not Copy" as the watermark. If you select this option, type the watermark text that you want in the Text box. Use the **Select Font** button to select the appropriate font settings.
- **Fill Type.** The watermark text or image can either be **Stretched** or **Tiled** to fill the target document or a defined document area.
- **Unit Type.** If a area is defined for the target document, the watermark is applied to the area rather than to the whole document.
- **Coordinates.** Specify the coordinates for where the watermark is applied. Click "**…**" to locate the image. Double-click to open the image file and identify the watermark coordinates, and then click **OK**.

Using Knowledge Package Loader to configure the Watermark component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Watermark component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Watermark component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Watermark component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Watermark component

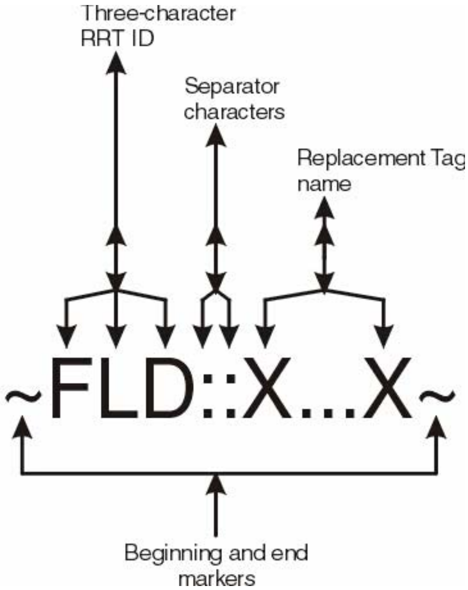
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Watermark component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Watermark component does not generate RRTs. However, all of the attributes can contain RRTs.

For example, when POP3 E-mail is the Capture component, "the Text box" parameter can be set to "**~POP::From~**" to watermark the images received by the value of the From field on the e-mail.

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

- The Watermark component does not support color watermarks.
- Only raster image files being processed and graphical watermarks are supported.
- The .PDF file format is not a supported file type.

VB/J Script component

Use the VB/J Script component as a Process or Route component with the most common scripting languages to manipulate and create your own custom-built capture program. You can write your scripts to gain access to external databases, manipulate internal files, or validate index data fields. You can also use the VB/J Script component to write scripts to gain access to external data sources, look up information that adds value to your capture process, and merge the necessary external data elements.

Scripts offer flexibility that helps you create custom-built Capture components quickly. As files come in, you can run a designated script to alter the files, and then perform various tasks such as saving the files to specific locations.

The VB/J Script component supports VBScript and JScript scripting languages.

Feature highlights

Choose a scripting language to create the VB/J Script component from the following two options:

- JScript
- VBScript

Use this component as a Process or Route component in any process.

The VB/J Script component can be used to process any file type.

Using the VB/J Script component

Use this component to perform the following tasks:

- Gain access to external database files from within your Capture task and validate captured data elements against your internal databases.
- Manipulate and reformat a file in the middle of a process to match your custom needs.
- Apply other program wrappers to files so that you can control the document format, security, and presentation.
- Add, delete, or modify field index data values to the process data space, reduce the amount manual data entry required, and increase your data throughput.
- Inform other users when a specific file type file has been received. You can run a script that sends an e-mail notification each time that particular file type is received.

NOTE When the VB/J Script component is the Route component in an AutoStore process, the script that you choose should provide the Route functionality.

Configuring the VB/J Script component

Use the appropriate procedure to open the **VB/J Script Configuration** dialog box to configure the VB/J Script component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the VB/J Script component.

The following attributes are available in the VB/JScript Configuration dialog box.

- **Name.** Type the name of the function that you want to run. For example, you might have many different _OnLoad functions in your script. You must specify which one you want to use. If you specified "Test" as the value of your **Name** field, then the Test_OnLoad function runs.
- **Language.** Select JScript or VBScript from the drop-down list.
- **Script.** Click "**…**" to browse for the script that you want to run. The script must be available in the same directory at runtime.

Using Knowledge Package Loader to configure the VB/J Script component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the VB/J Script component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the VB/J Script component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the VB/J Script component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the VB/J Script component

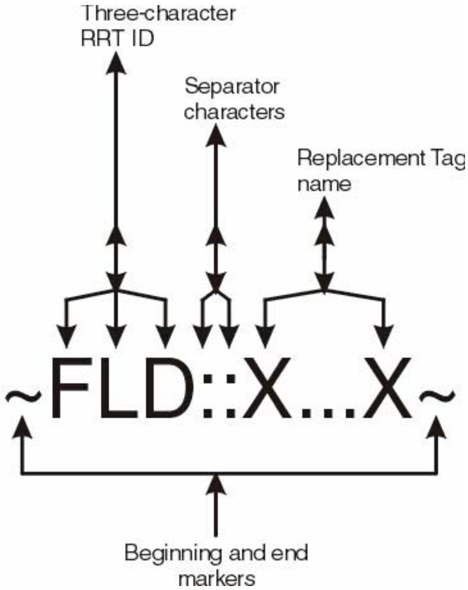
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the VB/J Script component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The VB/J Script component does not generate RRTs. However, all of the attributes can contain RRTs.

For example, when POP3 E-mail is the Capture component, set the "Record Type" parameter to "**~POP::Subject~**" to dynamically set the record type by using the subject field of the e-mail.

Troubleshooting tips

Problem description	**Solution**
Clicking **OK** does not close the dialog box.	Make sure that you have typed a name, selected a language, and specified a script in the configuration dialog box.
A VB/JScript error occurs.	Make sure that the file that you selected as your script is a valid file, and that it has a valid file extension.

Restrictions and limitations

- You must specify a name, language, and script in the **Configuration** dialog box.
- You must specify a valid script file.

7 Route components

AutoStore uses a Route component to store the data stream (images or data elements) into its final destination location. The Route component must always appear as the final component within a process.

Use Route components to store a file and related data elements into a final location. Each Route component is designed to work with a specific type of program, and will store data in that one program. You can store data in multiple programs by using a multiprocess chain design.

LAN Fax component

The LAN Fax component supports faxing capabilities that use a third-party LAN fax server. The LAN Fax component communicates with the network fax server by using a common directory folder and HP Scanfax technology. This folder can be a network folder if both the LAN fax component and the LAN fax server have read and write access to that component. Although not required, it is recommended that the LAN fax server be installed and the common folder established before the LAN Fax settings are configured.

Feature highlights

The LAN Fax component offers the following features:

- Send faxes by using a LAN fax server.
- Capture documents from remote sites by using any scanner, MFP, or digital sender, and route images to any of the supported fax servers.
- Create billing-account numbers for fax billing records.

NOTE The LAN Fax Route component does not support .PDF or .JPEG file formats.

Using the LAN Fax component

This component supports the following functions:

- **Copy-to-fax.** Connect all of your network-enabled digital senders to your enterprise fax servers.
- **Scan-to-fax.** Connect your desktop scanner directly to enterprise fax servers.
- **Email-to-fax.** Capture documents by using POP3 E-mail or SMTP Capture and route them to enterprise fax servers for processing.
- **Directory-to-fax.** Poll a directory and push each file into a fax server. Note that if you use the Poll Directory Capture component, you must use the Data Interchange Process component immediately after the Capture component.
- **XML-to-fax.** Receive AutoStore XML documents and send by fax.
- **Use VBScript.** Use VBScript to look up fax address books and integrate with your back-end fax database address books.

NOTE

If you are designing a process that captures files from the Poll Directory component and routes them to the LAN Fax component, make sure that you use a Data Interchange process component after the Poll Directory component. The Data Interchange component allows you to use RRTs for specifying field values that refer to phone numbers and other necessary data.

Case 1. If you design a process by using the Poll Directory Capture component, any Process component, and then the LAN Fax Route component, RRTs are not allowed for the LAN Fax component attribute fields (such as name, phone, and so on).

Case 2. If you design a process by using any blocking component as the Capture component, followed by any Process component, and then the LAN Fax component as the Route component, RRTs are enabled because the blocking component creates the fields.

Case 3. If you design a process by using the Poll Directory Capture component, followed by the Data Interchange Process component, followed by any Process components, and the LAN Fax Route component, the Data Interchange component creates the ability to set field values (for example name, phone, and so on) by using RRTs.

Licensing the LAN Fax component

Three types of licenses are available for this component: Evaluation, Licensed, and Expired.

- **Evaluation.** A 60-day fully functional component is available upon first installation.
- **Licensed.** The fully licensed component provides full capabilities indefinitely.
- **Expired.** After the evaluation period, unlicensed components expire without any further processing.

Configuring the LAN Fax component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the LAN Fax component.

The following attributes are available in the **LAN Fax** dialog box.

General tab

On this tab, specify basic information about the fax.

- **Fax Number.** Type the destination fax numbers, separated by commas.
- **Billing Account.** (optional) Type the billing code, if the fax server supports one. This code can be used for accounting purposes.
- **Description.** (optional) Type a description of the fax, if the fax server supports one.
- **Scanner Name.** (optional) Specify the scanner name from which the fax is originating, if the fax server supports one.
- **User Name.** (optional) To send a fax through a specific account on the LAN fax server, type a user name that exactly matches the LAN account on the server. Faxes that are sent without a user name are faxed from the LAN fax server default account on the LAN fax server.

NOTE

The availability of the attributes on the **Fax Settings** tab depends on your LAN fax server.

Fax Settings tab

Use this tab to set the attributes that are associated with how the fax is sent. Depending on the LAN fax server, some of the settings might not be available.

- **Error Correcting Mode.** Select this check box to activate error correction. Most fax machines have a feature called ECM (error correction mode), which permits them to continue transmission or reception despite minor (and usually intermittent) problems with the quality of the telephone line connection. When the quality of the telephone line connection is poor, transmission time increases as the two machines repeat data signals in an attempt to complete the transmission.
- **Retry.** Type a value between 1 and 99 if you want the LAN Fax component to retry sending the fax after a failed attempt. Type a 0 (zero) if you do not want the LAN Fax component to retry sending the fax after a failed attempt.
- **Interval.** If you type a value between 1 and 99 in the **Retry** text box, specify the number of minutes the LAN Fax component should wait between transmission attempts. Type a value between 1 and 60 minutes.
- **Resolution.** Select the quality of the fax by clicking a resolution value from the drop-down list.
- **Maximum Transmission Speed.** Specify the transmission speed by selecting a value from the drop-down list. This is the maximum baud rate at which you want the faxes to be transmitted. The LAN Fax product can also limit the transmission speed. Check your LAN fax server hardware and software specifications for recommended transmission speeds.

LAN Fax Inbox tab

Type the attributes related to where the fax documents are stored.

- **Folder Path.** Specify the path to the common folder where you want to place the fax data that will communicate with the LAN fax server. The folder can be a network folder if both the LAN Fax component and the LAN fax server software have read and write permissions to the folder. If the folder is a network folder, type the UNC (for example, `\\my_server\my_share_drive\my_share_folder`).

  Alternately, you can click the "**...**" button to browse for the folder name that you want.
- **Overwrite Existing File.** Select this check box to overwrite files that have the same name.
- **Rename File.** Check this box to rename the output file based on the Rename Schema settings.
- **Schema.** Type the schema name that is used to reformat the output file name. You can use Runtime Replacement Tags (RRT) to dynamically set the value of the schema.

NOTE Spaces are not valid in the file-naming schema.

Depending on which Capture component you are using, follow the appropriate procedure to open the LAN Fax dialog box and configure the LAN Fax component.

Using Knowledge Package Loader to configure the LAN Fax component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the LAN Fax component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the LAN Fax component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the LAN Fax component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the LAN Fax component

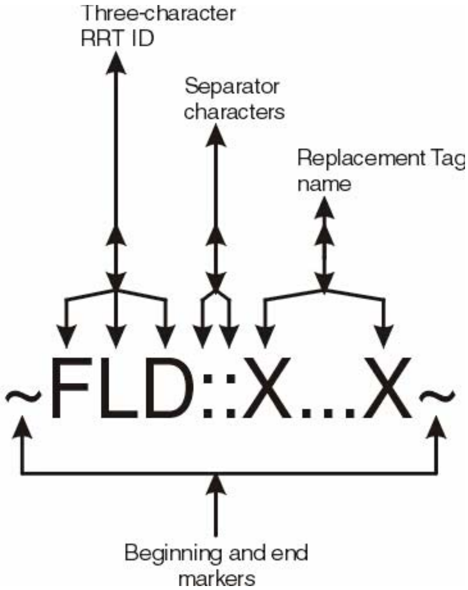
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the LAN Fax component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **LFX**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for the Schema field of this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names within a directory. The counter value concatenated with a name provides a unique file name.
FileExt	The original file-extension value.

The following example provides some sample usage for your reference:

~LFX::FileName~~LFX::Counter~ The value "Document5" is assigned if the original schema was "Document" and four (Document1 to Document4) files named "Document" were already within the destination folder path.

Field Replacement Tag Names

This component does not support FRTNs or the replacement of field names with metadata values.

Special Set Replacement Tag Names (SSRTN)

This component supports the Date/Time tag names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale

SSRTN	Description
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~LFX::%Y~-~LFX::%m~**" will be replaced by "**2004-9**"

Troubleshooting tips

Problem	Solution
A fax document could not be sent to the recipient.	Verify that the number you typed is a valid fax destination. Check the LAN fax server log for details.
The .HPF files are not written to the destination directory.	Make sure that you are not using mapped drives for your folder path. Mapped drives are not supported. Also make sure that the server has read and write permissions to the folder.

Restrictions and limitations

The LAN fax server validates the fax numbers. The LAN Fax component **does not** notify you if you attempt to fax to an invalid fax number; however, the LAN fax server activity log does.

LAN Fax supports a large number of the LAN Fax Server products that are currently available.

The LAN Fax Route component does not support .PDF or .JPEG file formats.

Folder Store component

The Folder Store component is a storage component that can be used to copy files into any local or network directory. Using directories is one of the fastest methods of implementing a document storage system. Storing files into flat folders requires no database and no software program, and results in lower costs. Use Folder Store to create business rules for folder location, security access, and file naming.

The Folder Store component can also check your security access before storing documents into folders. When the Check User Security option is activated, you can store documents in a destination directory only if the administrator has assigned you an appropriate access level. By using this feature, an organization can create a secured storage location based on security settings.

If you enable the Check User Security option, you must have Active Directory Services with a minimum of Windows 2000 and Service Pack 3 and later.

Feature highlights

You can use the Folder Store component to accomplish the following tasks:

- Create dynamic folder names and locations.
- Rename the scanned file based on the document index information, field tags, or Runtime Replacement Tags (RRTs).
- Check the user security level against the destination folder location. Only users who have write permission can store files into a destination location.

The Folder Store component is fully integrated with other Process components such as the Knowledge Package Builder, ABBYY FineReader, or Send to Printer components. All types of files, including images, can be processed through this component.

Using the Folder Store component

The following are two common scenarios for using the Folder Store component:

- You log onto a device, scan a document, and place the document in your directory on a shared network drive.
- You store information in a folder where another process can read and use the information.

To use the Folder Store component with an MFP

The following procedure is an example of how you can configure the Folder Store component for use with an MFP.

1. In the AutoStore Process Designer, on the toolbar click **File**, and then click **New**. In the **New** dialog box, click **Blank Process**, and then click **OK**.
2. In the **Process Information** dialog box, type a name for the process, and then click **OK**.
3. Click the **Capture** tab and drag the MFP (4100/9000) component into the new process.

4. Click the **Route** tab and drag the Folder Store component into the new process.

5. Double-click the MFP (4100/9000) component to gain access to configuration dialog box. The **Common MFP Group** appears. Unless you create another group, all MFPs are contained in the Common MFP Group and inherit the menus created for this group.

6. Click **Add Form**. On the **General** tab, type a name for the form (for example, `test form`). Select the scanning mode and file format, and then type the action button name (for example, `Select to Send Document`).

7. Click the **Components** tab, and the click "**...**" to browse for a path name.

8. Select the folder path and select the **Overwrite Existing File** check box.

9. Click **OK** to close the form.

10. Click the **SMTP Gateway** tab and type the hostname or IP address of at least one SMTP gateway. Note that because AutoStore uses the SMTP protocol to turn on the MFP component, this gateway is used to route e-mails that are sent from the MFP to the e-mail destinations.

11. Click the **Preferences** tab and type information for the following working directories: Home Directory, Processed Files, Rejected Files, and Rejected Emails. The Port should be set to **3232**.

12. Click **OK** to close the MFP component configuration.

Configuring the Folder Store component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Folder Store component.

The following attributes are available in the **Directory Store** dialog box:

- **Folder Path.** Use the text box to identify the destination folder path. The AutoStore server must have write permissions to the folder that you identify. The destination folder path can be dynamically set by the Capture component. For example, **C:\Publish \~ApplicationTag~** will use the application tag filed by the Digital Sender component to create dynamic folders.

- **Overwrite Existing File.** If this attribute is set to Yes, the program overwrites files that have the same name.

- **Rename File.** If this check box is selected, the program renames the output file based on the Rename Schema settings.

- **Schema.** Use this text box to set the schema that is used to reformat the output file name. You can use Runtime Replacement Tags (RRTs) to set the value of the schema dynamically.

- **Check User Security.** If this attribute is set to Yes, AutoStore checks for the write permission availability on the destination folder for the user who is specified in the **User Name** field. The system checks the user or sender security level to verify write permissions. The system references the Active Directory Services to check the security rights.

  The LocalSystem account in the AutoStore Service Manager can be replaced by DOMAIN\username with the user name password. This allows any AutoStore process to write to any shared network folder that the account owner (Domain\username) used to write to the shared network folder.

- **User Name.** Use this text box to type the user name for the directory. The most common use includes using RRT variables to replace the **User Name** field.

Depending on which Capture component you are using, follow the appropriate procedure to open the **Directory Store** dialog box and configure the Folder Store component.

Using Knowledge Package Loader to configure the Folder Store component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Folder Store component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Folder Store component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Folder Store component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Folder Store component

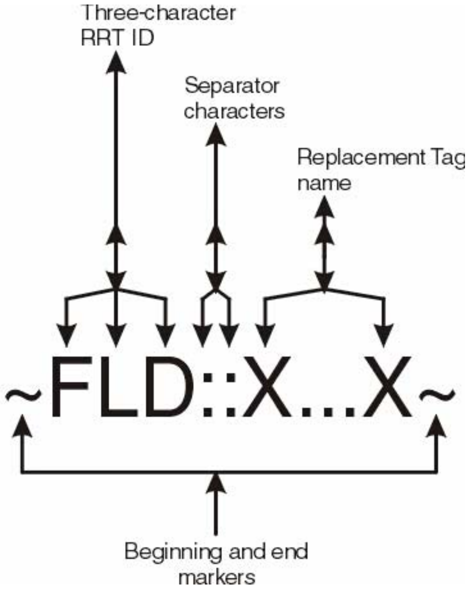
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Folder Store component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FLD**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	The original file-name value.
Counter	An incremental counter that is based on the duplicate filenames within a directory. The counter value concatenated with a name provides a unique file name.
FileExt	The original file-extension value.

The following is an example of the RRTN process:

~FLD::FileName~~FLD::Counter~

The value "Document5" is assigned if the original file name was "Document" and four (Document1 to Document4) files named "Document" were already within the destination folder path.

- **Field Replacement Tag Name** The FLD component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.
- **Special set replacement tag name (SSRTN).** The FLD component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)

SSRTN	Description
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"~FLD::%Y~-~FLD::%m~" is replaced with "2003-9"

Troubleshooting tips

Problem	Solution
Access to a directory is denied.	You do not have the appropriate permissions to the directory. Check with the administrator to get the appropriate permissions.
A "Path not found" message appears.	Drives are user-based. Make sure that the user name that is used for mapping drives is the same as the user name that runs the service. Click **Settings**, click **Control Panel**, and then click **AutoStore Service Manager** to view the user name that is logged into Windows. Confirm that this is the same user who mapped the drives.

Restrictions and limitations

- Use the **~FLD::Counter~** RRT in the schema name definition, instead of "**%c**", when configuring the Folder Store component.

FTP Store component

The FTP Store component is a destination component that provides the mechanism for storing files that are fed from a Capture component to the FTP site for storage. FTP Store also provides additional control over the FTP site transmission through support for secured FTP sites as well as bandwidth flow-control.

You can design your process to accommodate your business process. Any client (anonymous or specific) can then open an FTP session to this server to retrieve documents and accomplish other tasks that are necessary in the business process.

Feature highlights

The FTP Store component offers the following features:

- Active flow control
- Secured FTP site access
- Dynamic file renaming and storage

Using the FTP Store component

The following are examples of how the FTP Store component can be used.

- The FTP Store component can be used in an enterprise where .PDF documents from several sources have to be available on an FTP server. These documents can then be processed for printing or to send out as e-mail.
- Connect remote offices to a central office by using the FTP component. Scanners from all remote sites can easily be connected to a central site by using a secured FTP protocol.
- Use the bandwidth control feature of the FTP component to control the bandwidth usage when connecting remote scanning offices. Bandwidth control lowers the negative impact that distributed scanning can create in a transmission infrastructure.

Configuring the FTP Store component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the FTP Store component.

The following attributes are available in the **Directory Store** dialog box.

- **Server name.** Specify the IP address or the server name of the FTP site. You can also type the FTP site as `ftp://company-site.com`.
- **User name.** Type the user name of the client who will have permissions to the FTP site. After you type the user name, type the password that is associated with this user. If you do not type the user name, the permissions are set to anonymous by default.
- **User password.** Type the password that is associated with the user name.

- **Destination folder path.** You can type the folder where the documents will be stored on the FTP server. If you type a new folder in the dialog box, a new folder with the same name is created on the FTP server. However, if you do not type a path or folder name, the data is stored on the root directory of the FTP server.
- **Overwrite destination file if file exists.** Select this check box if you want the newly created files on the FTP site to overwrite the existing files that have the same name. If you do not select this option, the file will automatically be appended with a number that is incremented each time a newer version of a file with the same name is stored on the FTP site.
- **Rename File.** Select this check box if you want the output file to be renamed based on the Rename Schema settings.
- **Schema.** Type the Schema name for the output file name. You can use Runtime Replacement Tags (RRTs) to dynamically set the value of the schema.

Advanced setting

Click the **Advanced** button in the **FTP Store** dialog box to gain access to an easy method to control the speed at which the data is transmitted, called flow control. The purpose of the flow-control option is to manage the negative impact FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. Additional control is provided to test the flow-control speed and to provide an easy-to-use method for checking the transmission speed.

NOTE To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits dummy data to the FTP site to test the transmission bandwidth.

Use the controls that appear to adjust the following parameters on the FTP:

- **Activate Flow Control.** Select this check box to control the speed at which the data is transmitted. The purpose of the flow-control option is to manage the negative impact that FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. Additional control is provided to test the flow control speed and to provide an easy-to-use method for checking the transmission speed.

  Select the Activate Flow Control check box to activate the flow-control options.

  - **Transfer Rate in Milliseconds.** Use this control to set your preferred transfer intervals in milliseconds. The longer the interval, the slower the transmission rate.
  - **Buffer Size in Bytes.** Use this control to set the size of each transmitted buffer. The larger the buffer size, the higher the transmission rate and the larger the impact on the bandwidth.
- **Run Test.** The test is conducted against your FTP site. Make sure that you have specified the correct FTP site, user name, and password. The tests are conducted using 100 Kbyte buffers, and at the end of each test cycle the results are reported in 100 Kbytes/X second, where X is the number of seconds it takes to send 100 Kbytes. Normal black-and-white documents are between 25 to 50 Kbytes. The size of images can vary depending on the type of scanner settings, and you should observe your scanner setting parameters. Note that you must have write permissions to conduct FTP testing against a site.
- **Stop.** Use the **Stop** button to terminate the test-buffer transmission.

NOTE

To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits “dummy” data to the FTP site to test transmission bandwidth.

Depending on which Capture component you are using, follow the appropriate procedure to open the **FTP Store** dialog box and configure the FTP Store component.

Using Knowledge Package Loader to configure the FTP Store component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the FTP Store component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the FTP Store component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the FTP Store component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the FTP Store component

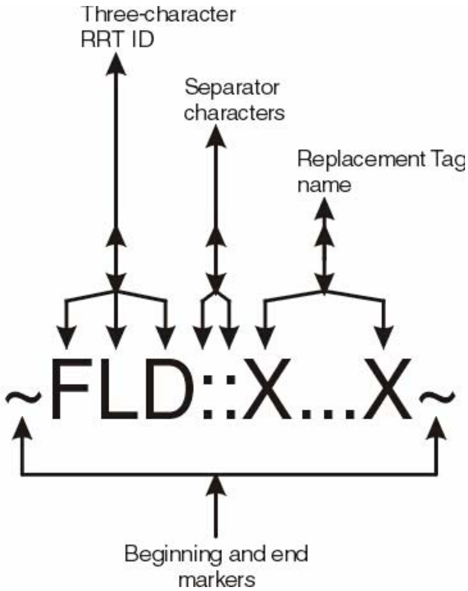
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the FTP Store component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FTP**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	**Description**
FileName	The original file-name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value concatenated with a name provides a unique file name.
FileExt	The original file-extension value.

The following is an example of the RRTN process:

~FTP::FileName~~FTP::Counter~

The value "Document5" is assigned if the original file name was "Document" and four (Document1 to Document4) files named "Document" were already within the destination folder path.

- **Field Replacement Tag Name** The FTP component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.
- **Special Set Replacement Tag Name** The FTP component supports the Date and Time special set replacement tag names (SSRTNs) shown in the following table:

SSRTN	**Description**
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name

SSRTN	Description
%B	The full month name
%d	The day of month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"~FTP::%Y~-~FTP::%m~" is replaced with "2003-9"

Troubleshooting tips

Problem	Solution
The "Access denied" message appears.	Check the user name and password to make sure that they are valid. Check the permissions for the user. Check to make sure that no firewalls are blocking the user's ability to gain access.

Restrictions and limitations

This component has no known restrictions or limitations at this time.

ABM Importer component

Use the ABM Importer component to distribute address book entries among groups of digital senders. You can create device groups and automate administrative tasks that are related to managing and maintaining the address books within each device.

Use the ABM Importer component with the ABM Exporter component to create a primary and secondary device configuration. Configure the ABM Exporter component to extract address-book entries from a single "primary" device, and create a "secondary" device group by using the ABM Importer component. When you run this process, all of your address-book entries will be exported from the primary device and imported into the secondary devices. This is the most efficient way to synchronize your digital sender address-book entries across an organization. When the two components are set up, an administrator needs to maintain the address-book entries and updates on the primary device so that they are automatically propagated to all secondary devices.

Feature highlights

The ABM Importer component provides the following features.

- Device group configuration support
- Single or multiple group support
- Address-book database support, including public e-mail, user profiles, public fax, printers, function keys, and prompt fields
- Selective address-book update support (allows you to select a subset of address-book types)

Using the ABM Importer component

Use the ABM Importer component to create groups of digital sender devices and send address-book updates in batch mode. To create the address-book batch file, you must use the ABM Exporter component. The ABM Exporter component allows you to export entries from a digital sender and store them into a batch file.

The ABM Importer can receive input directly from ABM Exporter or from a Poll Directory component. When using Poll Directory as a Capture component, place both of the batch files into the Inbox directory of the Poll Directory component. Note that all device database entries are cleared before a new batch file entry load begins.

Use the ABM Importer component to accomplish these tasks:

- Create groups of secondary devices.
- Keep backup files of your address-book entries in batch file format (using the ABM Exporter component) and restore address book entries.
- Perform mass updates across all of your digital sender devices.
- Create primary and secondary device relationships using the ABM Exporter and ABM Importer component pairs within the same process.
- Manage multiple device groups.

- Propagate your manual entries from one device to many devices.
- Support both older versions and new releases of firmware, including the prompt field definitions.

Configuring the ABM Importer component

The following attributes are available in the **ABM Importer** configuration dialog box:

- **NA Group.** Type the host name or the device IP address of the primary device from which you want to export address book entries. Click **Add** to define a new network appliance group. Each group has its own member devices. The device members should be mutually exclusive and not overlapping. If a device is a member of two groups, then that device will receive the same batch twice. Click **Rename** to rename an existing NA group. To remove an NA group, select a group name from the drop-down list, and then Click **Delete**.
- **Password.** Type in the administrator password for the digital sender devices. Note that all of the passwords for all of the devices within the same NA group must be the same. If you have devices that have different passwords, you must create different NA groups for each set of devices that share the same password.
- **Confirm Password.** Confirm that the password you typed in is correct by retyping it. This must match the **Password** field value.
- **Clear Address Books Prior to Import.** Select each check box to force a clear message into the digital sender device and clear its entries before loading the batch file entries.
  - **Public Email**
  - **User Profiles**
  - **Public Fax**
  - **Function Keys**
  - **Function Keys (new style)**
  - **Prompts Choices (new style)**
  - **Printers**
  - **Network Appliances**
  - **Name or IP Address**

The **Network Appliances** section of the dialog box provides an area for you to insert a device address into the current NA group by clicking **Insert**, or to remove an address by selecting an address, and then clicking **Delete**.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **ABM Importer** configuration dialog box and configure the ABM Importer component.

Using Knowledge Package Loader to configure the ABM Importer component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.

3. In the **Component Name** window, select the ABM Importer component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the ABM Importer component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the ABM Importer component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the ABM Importer component

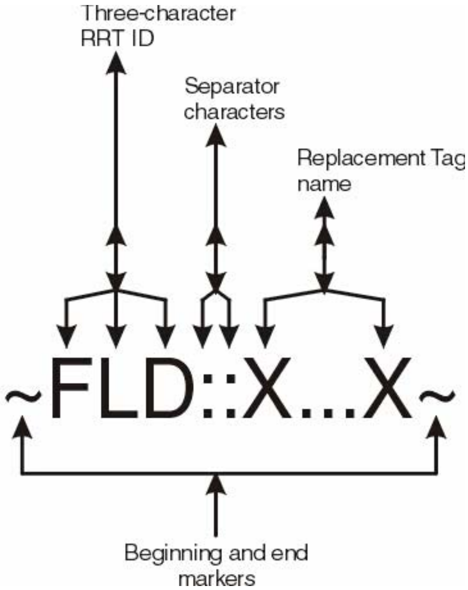
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the ABM Importer component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The ABM Importer component does not generate RRTs. However, all of the attributes can contain RRTs.

Troubleshooting tips

Problem	Solution
The devices begin to accumulate entries, rather than receiving an exact copy of the batch file.	You must select the appropriate check boxes in the configuration dialog box to clear the address-book entries. Make sure that the correct check boxes are selected and resend the batch file.
Some devices do not receive any updates.	The devices that do not receive updated entries after the ABM Importer component runs are either using a different administrator password or cannot be accessed during runtime. Another possibility is that the devices have a very old version of the firmware. Make sure that the digital sender devices have the most up-to-date version of the firmware.

Restrictions and limitations

- You can only send batch files to the digital senders.
- The only component that can generate batch files automatically is the ABM Exporter component. You can use the ABM Exporter component to export the address-book entries and store them into a batch file as a back up, and then use the ABM Importer component to restore the pre-existing batch files.

Send to Printer component

Use the Send to Printer component to send scanned images directly to a printer. You can also use this component to select the printer size, source, and zoom level. All of the print drivers for each printer must be installed and configured on the server.

Use the Send to Printer component to define and designate a printer as the destination for images that are sent to the server. This component enables the MFPs to define copying function keys that scan documents and automatically send them to a defined printer location.

Use the Send to Printer component to batch-print commonly used image-file formats such as .TIF and .JPG. You can also use the system to print in .PDF file format when you acquire the appropriate license.

Use the Send to Printer component to print documents to any printer that is configured on the computer where the system is running. The system can also switch between printers based on the name or the extension of the document file name. The Send to Printer component can be used both as a Process component and a Route component in a process workflow, depending on your needs.

The Send to Printer component manipulates different file formats without using any native program in the service computer. Because the printing process uses the printer drivers that are installed in the service computer, you must set up before configuring any printers that you want the component to use.

Feature highlights

Many business processes that use image documents incorporate printing as part of their workflow. The Send to Printer component satisfies many of the printing needs in a document-management environment. Whether obtaining a printed version of a document is the final goal of the process, or the system relies on printed versions as an alternate source of information, the Send to Printer component can be used to accommodate many scenarios where unattended printing is necessary.

Use the Send to Printer component to specify parameters that are common to most print drivers, such as paper size, printer name, pages per sheet, and orientation. Other parameters that are specific to a printer must be set in the local printer configuration.

The Send to Printer component has restricted capabilities for the type of files that can be printed. Only non-interactive printer drivers are supported. Some printer drivers might not be supported when the system is running as a service, because some printer drivers require an interactive response from you before printing.

Using the Send to Printer component

To use the Send to Printer component, you must first decide whether sending to a printer is the final step of the workflow or an intermediate step. This defines whether Send to Printer is used in the configuration as a Process component or as a Route component.

Use the Send to Printer component to accomplish these tasks:

- **Remote copy.** Scan files and route them to a remote printer.
- **File type based printing.** Scan files and send them to printers according to file types.

- **Color copy.** Scan in color and send the images to a color printer.
- **Print broadcast.** Create process chains with the Poll Directory Capture component and the Send to Printer Route component to broadcast multiple copies of a document to multiple printers.
- **Special printer features.** Accomplish automatic stapling and other specific printer functions by creating a copy of the printer driver that has the option for the function activated by default, and then using that specific printer driver.
- **Printing confirmation page.** Use **File Options** within your workflow process to store documents into a Success or Failure directory. If you want a confirmation page to report success or failure, use the Poll directory with the Send to Printer component to route files from the Success or Failure directory to a printer.

Licensing the Send to Printer component

The following types of licensing are available for this component:

- **Image Files.** The Send to Printer component supports all of the file formats described in the Restriction and limitations section, except for .PDF file formats.
- **Image and .PDF Files.** The Send to Printer component supports all of the file formats described in the Restrictions and limitations section, including .PDF file formats.

NOTE In the License Manager, you must turn on the PDF Enhanced level of licensing in addition to the base-level license if you want the AutoStore software to support .PDF file formats.

Configuring the Send to Printer component

Use the Send to Printer component to route images to printers that are located anywhere on the network. This component provides remote copying capabilities. Each page (tab) in the Send to Printer component represents a printer that can receive image files. The **General** tab defines the default printer, which is used for general-purpose print jobs. Based on the file extension, you can further determine the routing. For example, you can add a tab and define a separate printer for .TIF images, or define another printer for .PDF files.

The following attributes are available in the **Send to Printer Configuration** dialog box.

- **Activate.** This activates the Send to Printer component to send images to the printer.
- **Printer Name.** Use this drop-down list box to select the MFP.
- **Paper.** Use this drop-down list box to select the size of the paper. The values that appear change in relation to the printer and print driver programs. **Paper Source** indicates the paper source based on paper sources that are available on the printer driver. Select the **Auto Select** option if you want the printer to use a tray that supports the paper size that you selected.
- **Number of copies.** Type the number of copies that you want to print. To print a complete copy of the document before the first page of the next copy is printed, select the **Collate** check box.
- **Orientation.** Use these options to select the print-media orientation for printing.
- **Zoom.** Use this drop-down list box to set the number of pages that appear on each printed page.

- **Add.** Use this button to add a printer tab for a defined file extension. All files that match the file extension are routed to the printer that is defined on this tab rather than to the **General** tab.
- **Remove.** Use this button to remove a tab (other than the **General** tab). You must select a tab and then click **Remove** to delete the selected tab.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **Send to Printer Configuration** dialog box and configure the Send to Printer component.

Using Knowledge Package Loader to configure the Send to Printer component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Printer component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Printer component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Printer component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Printer component

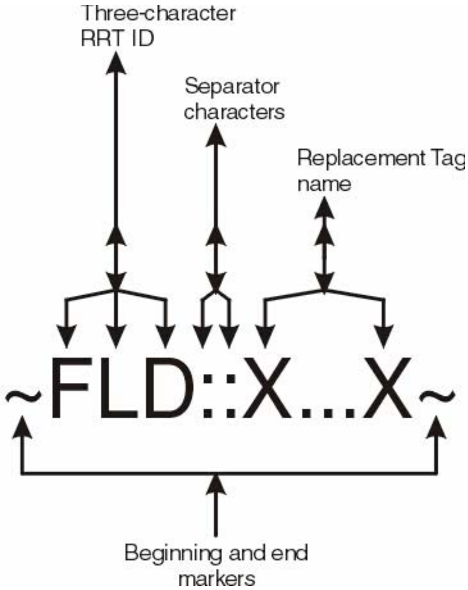
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, Select the Send to Printer component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Send to Printer component does not generate RRTs. However, all of the attributes can contain RRTs.

Troubleshooting tips

Problem	Solution
The server stops processing when you are trying to print document.	The print driver is trying to get interactive information. Make sure that the printer configuration does not request any information from you. For example, if the printer keeps a copy of the printing job, make sure that the file name is generated by the printer driver, rather than being requested from you. If you are using the system as a service, any kind of interactive actions cause the service to stop until a response is given. **NOTE** When you are using the interactive version of the system, the printer might need to request additional information from you.
An error occurs when you are trying to print a .PDF file, even though the system is licensed for .PDF file printing.	The .PDF file format that you are trying to use is not supported. If the program that generates the .PDF document can generate one of the supported .PDF file formats, use this instead (if your business process accepts the format).

Restrictions and limitations

Only non-interactive printer drivers are supported.

The following file formats are supported for Send to Printer.

JPEG formats

- JPEG File Interchange Format.
- Tagged Image File with JPEG compression.
- JPEG 2000 Format. This file format contains image data and extra information about the contents and organization of the file.

GIF formats

- CompuServe GIF.

TIFF formats

- Tagged Image File Format, with no compression and with RGB color space and 8-bit grayscale.
- Tagged Image File, with no compression and with CMYK color space.
- Tagged Image File, with no compression and with YCbCr color space.
- Tagged Image File with PackBits Compression and RGB color space.
- Tagged Image File with PackBits Compression and CMYK color space.
- Tagged Image File with PackBits Compression and color YCbCr space.
- Tagged Image File with CMP Compression.
- Tagged Image File with JBIG Compression.
- Tagged Image File with a vector image saved as a .DXF.
- Tagged Image File with JPEG 2000 Compression. This file format contains only a stream of image data.
- Tagged Image File with Wavelet CMP Compression.

BMP formats

- Windows .BMP, with no compression.
- Windows .BMP, with RLE compression.
- OS/2 .BMP version 1.*x*.
- OS/2 .BMP version 2.*x*.
- Wireless Bitmap file. Type 0.

WMF and EMF formats

- Windows MetaFile.
- Windows Enhanced MetaFile.

Exif formats

- Exif file containing a TIFF image, with no compression and with RGB color space.
- Exif file containing a TIFF image, with no compression and with YCbCr color space.
- Exif file containing a JPEG compressed image.

1-Bit FAX formats

- TIFF, compressed by using CCITT.
- TIFF, compressed by using CCITT, group 3, 1 dimension.
- TIFF, compressed by using CCITT, group 3, 2 dimensions.
- TIFF, compressed by using CCITT, group 4.
- Raw FAX, compressed by using CCITT group 3, 1 dimension.
- Raw FAX, compressed by using CCITT group 3, 2 dimensions.
- Raw FAX, compressed by using CCITT group 4.
- IOCA, compressed by using CCITT group 3, 1 dimension.
- IOCA, compressed by using CCITT group 3, 2 dimensions.
- IOCA, compressed by using CCITT group 4.
- IOCA, compressed by using IBM MMR, with the MO:DCA wrapper.
- IOCA, uncompressed, with the MO:DCA wrapper.

Other 1-Bit formats

- MacPaint.
- Portable Bitmap - ASCII File.
- Portable Bitmap - Binary File.
- XBitmap File.
- Microsoft Paint.

PDF formats (Image and .PDF Files license required)

- PDF 1.3.

NOTE In the License Manager, you must turn on the PDF Enhanced level of licensing in addition to the base-level license if you want the AutoStore software to support .PDF file formats.

Other formats

- PostScript 3 files (Image and .PDF Files license is required).
- EPS (Image and .PDF Files license is required).

Send to Mail Recipient component

The Send to Mail Recipient component provides standard SMTP messaging capabilities. Use this component to send a processed file attachment to designated recipients as an e-mail.

Use this component to distribute files, metadata, and additional information by using e-mail. For example, you can use the Digital Sender component as your Capture component to scan documents, and then convert the scanned documents into .PDF files and e-mail them to designated recipients as an attachment.

Feature highlights

Use the Send to Mail Recipient component to perform the following tasks:

- Attach processed files to an e-mail, and then send the e-mail to a designated recipient(s).
- Specify the appropriate SMTP gateway to be used for delivering the e-mail message.
- Return the processed files as an e-mail attachment to the sender (requires the sender's e-mail address).

Using the Send to Mail Recipient component

If you want to use a process in which you capture documents, process them, and send the processed documents as an e-mail attachment, set up the process by configuring the appropriate Capture and Process components, and then add the Send to Mail Recipient component as the Route component. Then configure the Send to Mail Recipient properties.

Set the Server Name to the SMTP gateway that you use to send e-mail. For example, use **webmail.company_name.com** or use the IP address of your SMTP mail server. Set the **From** field to the name of the person who is sending the e-mail. You can use the RRTs of other components to dynamically set the values for **From** field. For example, when your Capture component is POP3 e-mail and you want the processed file to be sent back to the sender, you set the **From** and the **To** field to **~POP::From~**. This person receives the processed documents by simply sending document to a POP3 e-mail account.

The **Subject** and **Body** fields are optional. The subject indicates the subject for an e-mail that you send. The body of the e-mail is anything additional that you want to add to the e-mail. Keep in mind that you can use RRTs of other components in any fields to create custom and dynamic values.

File input types are not restricted for this component. Any file type can be processed through this component.

Configuring the Send to Mail Recipient component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Send to Mail Recipient component.

The following attributes are available in the **Send to Mail Recipient** dialog box:

- **Server Name.** (required) Type the server IP address or the SMTP gateway host name through which you send your e-mail.
- **From.** (required) Type the sender's e-mail address.
- **To.** (required) Type the recipient's e-mail address. Separate multiple recipient addresses with semicolons.
- **Subject.** (optional) Specify the subject of the e-mail.
- **Body.** (optional) Provide content for the e-mail.

Depending on which Capture component you are using, follow the appropriate procedure to open the **Send to Mail Recipient** dialog box and configure the Send to Mail Recipient component.

Using Knowledge Package Loader to configure the Send to Mail Recipient component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Mail Recipient component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Mail Recipient component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Mail Recipient component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Mail Recipient component

1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to Mail Recipient component.
5. Click **Configure**.

Configuration example

Here is an example of capturing documents, processing them, and sending them as an e-mail attachment. Set up the process by combining the correct Capture, Process, and Send to Mail Recipient Route components. Then, set up the Send to Mail Recipient properties.

Set the **Server Name** to the SMTP gateway used for sending e-mail. For example, use **webmail.company_name.com** or use the IP address of your SMTP mail server.

Set the **From** field to who is sending this e-mail. You can use the RRTs of other components to dynamically set the values for the **From** field. For example, when your Capture component is a POP3 E-mail and you like to have the processed file sent back to the sender, then set the **From** and the **To** field to ~POP::From~. This person would receive processed documents back by sending the document to a POP3 e-mail account.

Subject and **Body** are optional fields. The subject should briefly describe the content of the e-mail. The body of the e-mail should be the actual content that you want to send. You can use RRTs of other components in any fields to create custom and dynamic values.

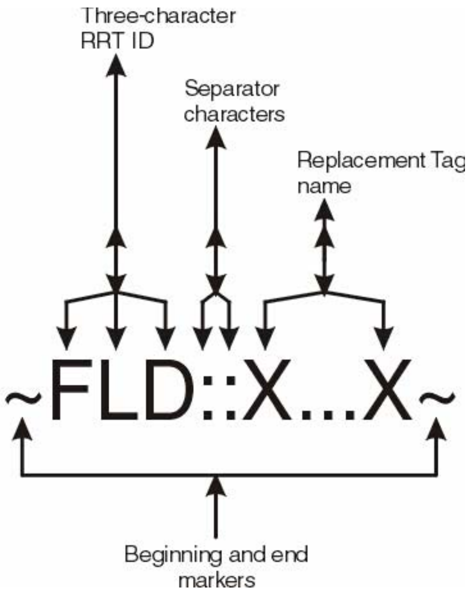
No restrictions exist for the file input types of this component. Any file type can be processed through the Send to Mail Recipient component.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Send to Mail Recipient component does not generate RRTs; however, all of the attributes can contain RRTs. For example, when POP3 E-mail is the Capture component, the **To** attribute can be set to "**~POP::To~**".

Troubleshooting tips

Problem	**Solution**
This component is not sending e-mail.	Make sure that your sever name and the recipient's e-mail address are correct. If you have multiple recipients, make sure that each is separated by using a semicolon.
The SMTP gateway server rejects the e-mails.	Make sure that the SMTP mail server allows relaying.
The e-mail addresses entered in the **To** field are not validated. Non-compliant addresses could be used.	Use the Notification Process component in your HP AutoStore process. Using the Notification component allows you to require a user name and password so that you can prevent non-compliant addresses from being used.

Restrictions and limitations

The **Server Name**, **From**, and **To** fields are mandatory fields. Be sure to enter valid values for each of these fields.

Send to PC component

Use the Send to PC component as a file transfer tool to connect people and devices. The Send to PC component allows devices such as scanners, network devices, and digital cameras to send files and related data directly to an individual's computer. You can also use the Send to PC component to manage file content connectivity, format, and security. This component can be used to deliver documents to a designated inbox on computers that are running the Send to PC client program.

The Send to PC component requires minimal set up and identification, which makes it easy to use. It is fully integrated with Windows 2000 and Windows XP. The Send to PC component maintains a list of NT user names for client PCs that are currently connected to the server. When processing a job, the Send to PC component will be assigned a list of users that the documents with a job should be delivered to. The assigned list will be compared with the maintained list to determine the location of the document delivery.

You can specify a secret key on the Send to PC client program. If that secret key is not specified, document delivery is prevented.

Feature Highlights

Use the Send to PC component to perform the following tasks:

- Deliver documents to a designated inbox on a client computer.
- Deliver documents to one or more client computers.
- Protect document delivery to designated client inboxes by defining a secret key on the client program. Documents will only be delivered if the secret key is provided.

This component works independently from the type of Capture component that is used in the process.

Using the Send to PC component

The following are examples of how to use the Send to PC component:

- **Send to My PC.** MFPs that have authentication turned on can be configured with a **Send to PC** button. When you press this button, images are scanned and then delivered to your authenticated designated inbox. This alleviates extra load and stress on the mail server.
- **Send to user workstation.** Any user can scan documents from an MFP and designate a user inbox for delivery. The recipient can be anywhere within the network.

NOTE When a user is logged into more than one computer at the same time, the last computer to broadcast that it is connected to the server will be the computer that receives a scanned document.

Configuring the Send to PC component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Send to PC component.

The following attributes are available in the **Send to PC** dialog box:

- **Send To.** (required) Type the names of users who should receive incoming documents. The user name should be the NT name, such as `DOMAIN\JOHNDOE`. Use commas to separate the names.
- **Advanced button.** Click the **Advanced** button to set the following connection attributes:
  - **Default Domain.** This is the domain name that will be attached to the user names in the **Send To** list that do not already have a domain name assigned to them.

    For example, if there is a user name JohnD on domain US1, and the default domain is US1, either user name `JohnD` or `US1\JohnD` can be entered into this field. In this case, the domain portion of the user name is not required. However, if there is a user JohnD on domain US2 as well, then the domain name and user name must be entered as `US2\JohnD`.
  - **Connection Options.** The following connection options are available:

    File Transmit Port. This port is used to deliver documents to client computers. The default value is **3711**.

    Registration Port. This port is used by client computers to register with this server component. The default value is **3711**.
  - **Separator.** The separator that is placed in between the user name and secret key. If secret keys are not being used, then entering the user name is sufficient. The secret key is an option on the Send to PC Client. If the Send to PC Clients have entered a value in the secret key input box, then to send a document to that client's PC, the secret key must be provided in the **Send To** input box of the Send to PC server component. Use the separator in between the user name and the secret key to separate them.

    It is recommended that you use "@" as the separator. The comma "," and semicolon ";" are invalid separators.

NOTE If you are running the server and client components on the same computer, the **File Transmit Port** and **Registration Port** should be set to different values.

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **Send to PC** dialog box and configure the Send to PC component.

Using Knowledge Package Loader to configure the Send to PC component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to PC component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to PC component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.

4. Click the **Components** tab.
5. In the **Name** column, select the Send to PC component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to PC component

1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to PC component.
5. Click **Configure**.

Send to PC Client

Set the Send to PC Client attributes with these settings.

Configuration

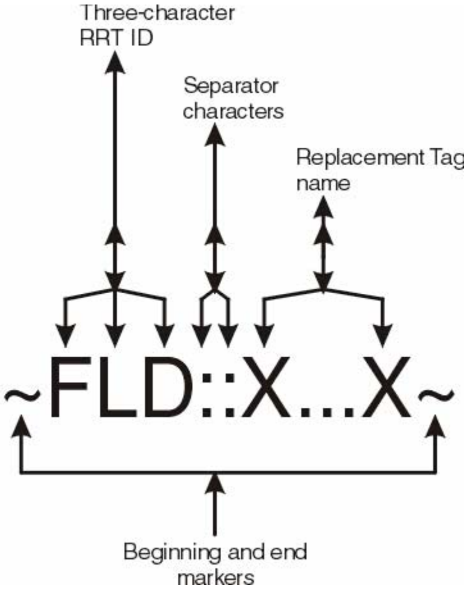
- **Server.** This is the name of the Send to PC server that delivers documents.
- **Test.** Use this button to check the validity of the server that is specified in the Server attribute.
- **User Name.** This is the name of the user that is currently logged on <NT Domain\User name>.
- **Secret Key.** This is the secret key that protects your inbox. If a secret key has been defined, then documents are delivered to this inbox only when the appropriate key is provided. The secret key is an option on the Send to PC Client. If the Send to PC Clients have entered a value in the secret key input box, then to send a document to that client's PC, the secret key must be provided in the **Send To** input box of the Send to PC server component.
- **Automatically start when Microsoft Windows starts.** As a convenience, the Send to PC client starts every time you turn on your computer. If you want to turn off this feature, click to clear the check box that appears next to this option.
- **Automatically connect at startup.** As a convenience, the Send to PC client connects automatically to the defined Send to PC server at startup. If you want to turn off this feature, click to clear the check mark that appears next to this option.
- **Directory for files received.** This is the directory where incoming documents are saved to. The path can be directly typed into the field, or you can use the Browse key, located on the right side of the field, to search for a path.
- **Connect.** Click this button to connect to the specified Send to PC server and register as client that is able to receive documents.
- **Disconnect.** Click this button to disconnect from the Send to PC server.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Send to PC component does not generate RRTs; however, all of the attributes can contain RRTs. For example, when POP3 E-mail is the Capture component, the **Send To** attribute can be set to "**~POP::To~**" to dynamically set the recipient inbox based on the **To** field of the e-mail.

Another example is using the MFP (4100/9000) as the Capture component and **~M94::%Sender%~@~M94::%secret key%~** where Sender **~M94::%Sender%~** is the Authenticated user name and **~M94::%secret key%~** is a field on an MFP form that the end user fills out.

Troubleshooting tips

Problem	Solution
The client cannot connect to a running Send to PC server.	Make sure that the client and server computers have Microsoft TCP/IP installed. If not, install the TCP/IP protocol. Make sure that client computer can ping the server computer (by using IP address) **ping <server_ip_address>** If you receive no reply, then your computer is probably configured incorrectly or a problem exists with the physical network connection of your computer. Make sure that the **Registration Port** and **File Transmit** ports on the server and client match. By default, the port(s) are set to 3711. Also, make sure that the ports are not blocked by a firewall. You can check by using Telnet at the command prompt to connect to the port 3711.

Restrictions and limitations

- Document metadata is not delivered to the client's computer.
- The client directory for received files cannot be dynamically set or changed at runtime.

Send to FTP

Use the Send to FTP component to store files that are fed from a Capture component to the FTP site for storage. The Send to FTP component is available as either an eConnector (Process) or a Route component. This component provides additional control over the FTP site transmission by providing support for secured FTP sites and bandwidth flow-control.

You can design your process to accommodate your business process. Any client (anonymous or specific) can then open an FTP session to this server to retrieve documents and accomplish other necessary tasks in the business process.

The Send to FTP Route and eConnector (Process) components are identical expect that the Route component only stores the files and does not pass the files on, because the Route component is the last component in the AutoStore process. The Send to FTP eConnector (Process) component passes the files on to the next component in the process. You can also use the Send to FTP eConnector component to enable or disable document pass-through. Enabling this option makes the document available for other components in the process.

Features

The Send to FTP component offers the following features.

- Configuration of multiple FTP sites
- Active flow control
- Secured FTP site access
- Dynamic file renaming and storage

Using the Send to FTP component

The Send to FTP component is very similar to the FTP Store component but has some added functionality. The following are examples of how the Send to FTP component can be used.

- The Send to FTP component can be used in an enterprise where .PDF documents from several sources have to be available on an FTP server. These documents can then be processed for printing or to send as e-mail.
- Connect remote offices to a central office by using Send to FTP component. Scanners from all remote sites can easily be connected to a central site by using a secured FTP protocol.
- Use the bandwidth-control feature of the FTP component to control the bandwidth usage when remote scanning offices are connected to the central office. Bandwidth control lowers the negative impact that distributed scanning can create in a transmission infrastructure.

Configuring the Send to FTP component

Set the attributes of the FTP sites by using the options on the Send to FTP component.

- **Pass-through documents.** Select this check box if you want the documents to be available to the other components in your process.

NOTE This option is available only on the Send to FTP eConnector (Process) component, not on the Send to FTP Route component.

- **Add.** Click **Add** to add FTP sites to your process.
- **Server.** Specify the IP address or the server name of the FTP site. You can also specify the FTP site as `ftp://companysite.com`.
- **User Name.** Type the user name of the client who will have permissions to the FTP site. After you type the user name, type the password that is associated with this user. If you do not type the user name, the permissions are set to anonymous by default.
- **User Password.** Type the password that is associated with the user name.
- **Folder Path.** You can type the folder path where the documents will be stored on the FTP server. If you specify a new folder name in the dialog box, a new folder with the same name is created on the FTP server. However, if you do not specify a path or folder name, the data is stored on the root directory of the FTP server.
- **Overwrite Existing.** Select this check box if you want the newly created files on the FTP site to overwrite the existing files that have the same name. If you do not select this option, the file will automatically be appended with a number that is incremented each time a newer version of a file with the same name is stored on the FTP site.
- **Rename File.** Select this check box if you want the output file to be renamed.
- **Schema.** Type the Schema name for the output file name. You can use Runtime Replacement Tags (RRTs) to dynamically set the value of the schema.

Advance setting

Click the **Advanced** button in the **Configurations** dialog box to gain access to flow control, an easy method of controlling the speed at which the data is transmitted. The flow-control option manages the negative impact that FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. You can also test the flow-control speed and check the transmission speed.

NOTE To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits "dummy" data to the FTP site in order to test the transmission bandwidth.

The FTP server must be configured so that the account that is used to transfer files has write permissions to the FTP server. Some operating systems also require the account to have write permissions to the destination folder. For example, the if you are using the MS NTSF file system, then the account used to transfer files must be given write permissions by selecting the appropriate Security settings for that folder. The account must also have the appropriate account access permissions configured in the FTP configuration option in IIS Administrator.

Use the controls that appear to adjust the following parameters on the FTP:

- **Activate Flow Control.** Select this check box to control the speed at which the data is transmitted. The flow-control option manages the negative impact that FTP transmission might have on the transmission bandwidth from remote sites. Use the flow-control knobs to control the size of the buffer and the transmission interval of each buffer. You can also test the flow-control speed and check the transmission speed.

  Select the Activate Flow Control check box to activate the flow-control options.

  - **Transfer Rate in Milliseconds.** Use this control to set your preferred transfer intervals in milliseconds. The longer the interval, the slower the transmission rate.
  - **Buffer Size in Bytes.** Use this control to set the size of each transmitted buffer. The larger the buffer size, the higher the transmission rate and the larger the impact on the bandwidth.
- **Run Test.** The test is conducted against your FTP site. Make sure that you have specified the correct FTP site, user name, and password. The tests are conducted using 100 Kb buffers, and at the end of each test cycle the results are reported in 100 Kb/X second, where X is the number of seconds it takes to send 100 Kb. Normal black-and-white documents are between 25 to 50 Kb. The size of images can vary depending on the type of scanner settings, and you should observe your scanner-setting parameters. Note that you must have write permissions to conduct FTP testing against a site.
- **Stop.** Use the **Stop** button to terminate the test-buffer transmission.

NOTE To use the test feature, you must have permissions to the FTP site from your computer. This feature establishes connection to the FTP site and transmits "dummy" data to the FTP site to test transmission bandwidth.

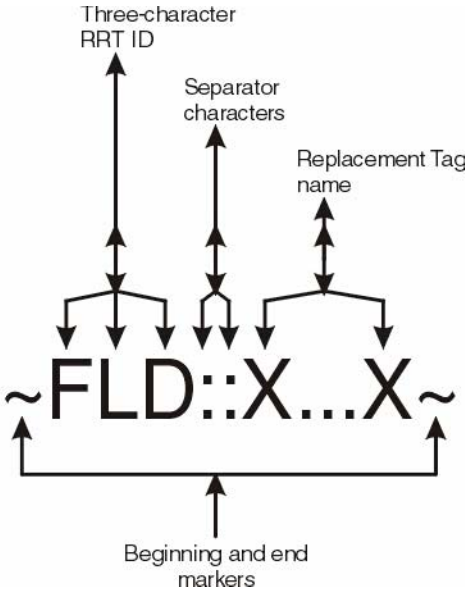
- **Remove.** Select the FTP site that you want to delete, and then click **Remove**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **FTP**.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for the **Schema** field of this component.

Name	Description
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value that is concatenated with a name provides a unique file name.
FileExt	This is the original file extension value.
Path	This is the folder path associated with an FTP server. For example, if you have configured three FTP servers or folder paths, then this RRTN can take the following values: Path1, Path2, and Path3. Path1 refers to the first path entry configured, Path2 refers to the second path entry configured, and so on.

The following is an example of the RRTN process:

~FTP::FileName~~FTP::Counter~

The value Document5 is assigned if the original file name was "Document" and four (Document1 do Document4) files named "Document" were already within the destination folder path.

NOTE

The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~FTP::FileName~**, **~FTP::Counter~**, or **~FTP::FileExt~** with any other component except the Send to FTP eConnector (Process) component, and it must be used with the **Rename** field.

NOTE

You can create and display the counter with the required number of leading spaces and leading zeroes. For example, if the file name is TEST.DOC, and the rename schema is **~FTP::FileName~~%03STF::Counter~~FTP::FileExt~**, then the resulting file names are TEST001.DOC, TEST002.DOC, and so on.

If the file name is TEST.DOC and the rename schema is **~FTP::FileName~~%3FTP::Counter~~FTP::FileExt~**, the resulting file names are TEST 1.DOC, TEST 2.DOC, and so on (note the two spaces after "TEST").

Field replacement tag name (FRTN). This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name

SSRTN	Description
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~FTP::%Y~-~FTP::%m~**" is replaced with "2004-10"

Troubleshooting

Problem	Solution
The files cannot be copied into the destination directory.	● Make sure that the destination FTP folder has write permissions. ● Make sure that the user name and password are valid. ● Make sure that the user has the appropriate permissions. ● Make sure that there are no firewalls. ● If **Overwrite Existing** is not selected, make sure that a file with the same name does not already exist.

Restrictions and limitations

This component has no known restrictions or limitations at this time.

Send to Database Route component

Use this component in a process to directly write images, files, or data into open database connectivity (ODBC) database tables such as Microsoft Access, or Visual Fox Pro.

Use this component to directly update structured query language (SQL) tables with images and data elements, or to create an archival system.

NOTE The ODBC data source must be set up correctly before you configure the Send to Database component. ODBC drivers must be set up before you use this component. Please refer to the ODBC driver provider for additional information about how to set up the ODBC drivers for your type of database.

Feature highlights

Use the Send to Database component to perform the following tasks:

- Select from various data sources where you can store data and binary documents.
- Provide the user name and password for access security.
- Select the table name (location) where you want to place the image and field values.
- Select the binary image field name (table column) into which you want to save an image.
- Add, modify, and remove field values to and from a database table.

Using the Send to Database Route component

The most common use of the Send to Database component is for archival purposes. You can store files, data, and documents in binary format in ODBC-compliant tables for long-term archival and record keeping. For example, you can use the MFP (4100/9000) as the Capture component in the APD, then use that component to configure the Send to Database component. Assign RRTs to store the images, files, and new field names into the data source that you select.

The Send to Database component can be used with any file or document type.

Configuring the Send to Database Route component

Use the appropriate procedure to open the **Send to Database** configuration dialog box to configure the Send to Database component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Send to Database component.

The following attributes are available in the **Send to Database** configuration dialog box.

- **General tab.** Enter the following data source attributes on this tab.
  - **Data Source.** Click "**…**" for a list of available data sources. Click to select a data source. This is a required field.
  - **User Name.** (optional) Type the user name to be used for logging in to the Data Source.
  - **Password.** (optional) Type the password to be used for logging in to the Data Source.
  - **Table Name.** Type the table name to which you want to connect. This is a required field.
  - **Image Field Name.** Select the field name (table column) in which you want to save an image. **Image Field Name** has to be a binary large object (BLOB) field or the process will fail when it is run. This field must be configured.

  If you want to create new fields in the database, proceed to the **Field Values** tab and select **Add**.
- **Field Values tab.** Use this tab to create, modify, or remove fields in the database.
  - **Add.** Click this button to add a new field value entries to the database table. The new field value can contain Runtime Replacement Tags (RRTs).
  - **Modify.** Click this button to modify the field value entries for the database table.
  - **Remove.** Click this button to remove a field value entries from the database table.

Using Knowledge Package Loader to configure the Send to Database (Route) component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Database component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Database (Route) component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Database component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Database (Route) component

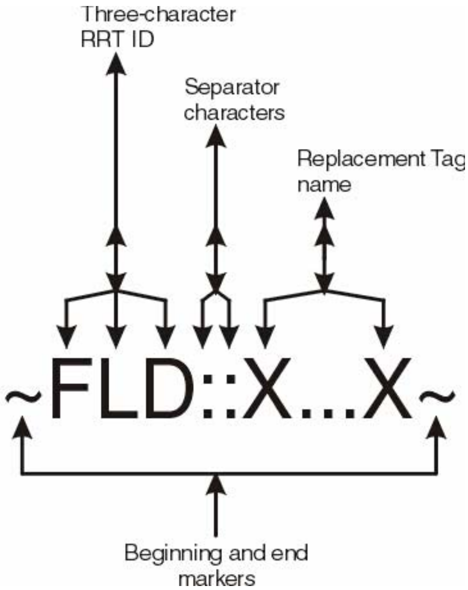
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to Database component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs, however all the parameters can contain RRT strings. For example, when POP3 E-mail is the Capture component, the "Data Source" parameter can be set to "**~POP::To~**" to dynamically set the Data Source name using the **To** field of the e-mail.

Troubleshooting tips

Problem	Solution
An error appears when you click the **Table** browse button.	You either do not have a valid user name and password, or you have an invalid Data Source. Make sure that the Data Source that you selected is actually connected and is configured correctly.
The "Duplicate output destination" error message appears.	This error occurs when your process has the ABBYY FormReader v6.0 component along with the Send to Database component in the following scenario: ● You add a field value entry in the Send to Database component. ● The **Export All Fields** box in ABBYY FormReader v6.0 component is selected. ● An identical field name (to the one specified in the Send to Database component) is exported to the database. To resolve this error, change the field value entry in the **Field Value** tab of the Send to Database component.

Restrictions and limitations

- If you need to create a database table, do so from within the database and create the tables from there. You cannot create tables dynamically from this component.
- The data source, table name, and the image-field name fields are mandatory and must be configured.
- The image-field name must be a binary large object (BLOB) field.
- All images within a BLOB field are stored as binary values. You must develop your own image-extraction tool in order to extract and view the images.

Send to Folder

Use the Send to Folder component to copy files to any local or network directory. Using directories is one of the fastest methods of implementing a document-storage system. Storing files into flat folders requires no database and no software program, and results in lower costs. Use the Send to Folder component to create business rules for folder location, security access, and file naming of scanned images and processed files.

The Send to Folder component can also check your security access before storing documents into folders. When the check security feature is activated, you can store documents in a destination directory only if the administrator has assigned you an appropriate access level. By using this feature, an organization can create a secured storage location based on security settings.

The Send to Folder Route and eConnector (Process) components are identical except that the Send to Folder Route component only stores the files and does not pass the files on to other components because the Route component is the last component in the process. The Send to Folder eConnector (Process) component passes the files on to the next component in the process. You can also use the Send to Folder eConnector component to enable or disable document pass-through. Enabling this option makes the document available for other components in the process.

NOTE This component requires Active Directory Services with minimum of NT 4.0.

Features

You can use the Send to Folder component to accomplish the following tasks:

- Create dynamic folder names and locations.
- Rename the scanned files based on the document index information, field tabs, or runtime replacement tags (RRTs).
- Check the user security level against the destination folder location. Only users who have write permission can store files into a destination location.
- Specify additional information about documents that are stored in the folder by using keywords and comments.

  The Send to Folder component is fully integrated with other components such as barcode, form recognition, OCR, PDF, and so on. For example, a barcode value can be used as part of a destination folder name to create a dynamic destination folder based on the barcode values that are on the document. All types of files, including images, can be processed through this component.

Using the Send to Folder component

The following are two common scenarios for using the Send to Folder component:

- You log onto a device, scan a document, and place the document in your directory on a shared network drive.
- You store information in a folder where another process can read and use the information.

Configuring the Send to Folder component

Use the options described here to define the storage structure of documents.

- **Pass-through documents.** Select this check box if you want the documents to be available to other components in your process.

NOTE This option is available only on the Send to Folder eConnector (Process) component, not on the Send to Folder Route component.

- **Add.** Click **Add** to add a folder path to your process.
- **Remove.** Select the folder path that you want to delete, and then click **Remove**.

General tab

Use the options on this tab to set the following attributes.

- **Folder Path.** Use the text box to identify the destination-folder path. The AutoStore server must have write authorization to the folder that you identify. The Capture component can dynamically set the destination folder path.
- **Overwrite Existing File.** If you select this check box, the program overwrites files that have the same name. If you do not select this check box and a file that has the same name exists in the destination-folder path, an error message is generated.
- **Rename File.** Select this check box if you want to rename the output file.
- **Schema.** Use this text box to set the schema that is used to reformat the output file name. You can use runtime replacement tags (RRTs) to set the value of the schema dynamically.
- **Check User Security.** If you select this check box, the software checks the user's or the sender's security level to verify the write access. This refers to a lookup in the user's Active Directory Services to find out if the user has security rights to copy the files to the designated folder.
- **User Name.** Type the name of the user for the directory.

Summary tab

Use the options on this tab to set the document attributes. These options appear when you select the file and right-click **Properties**.

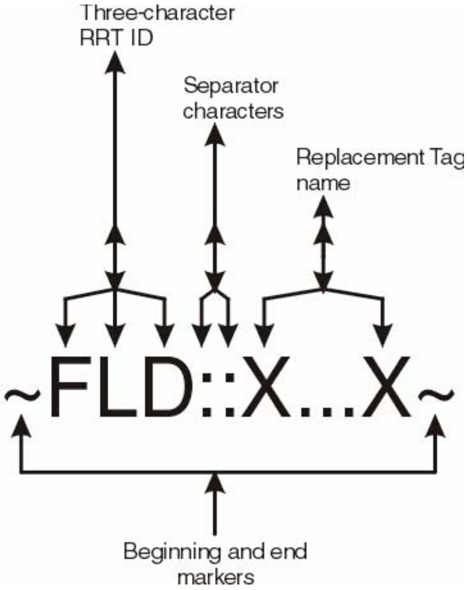
- **Subject.** Type a subject for your document.
- **Title.** Type the title of your document.
- **Author.** Specify the name of the author of your document.
- **Category.** Specify a category for your document.
- **Keywords.** Type keywords that are associated with your document. The keywords assist with future searches for the document. You can use a comma or a space to separate the keywords.
- **Comments.** Type comments that provide necessary information about your document.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **STF**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for the **Schema** field of this component.

Name	Description
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value that is concatenated with a name provides a unique file name.
FileExt	This is the original file extension value.
Path	This is the folder path entry. For example, if you have configured three folder paths, then this RRTN can take the following values: Path1, Path2, and Path3. Path1 refers to the first path entry configured, Path2 refers to the second path entry configured, and so on.

The following is an example of the RRTN process:

~STF::FileName~~STF::Counter~

The value Document5 is assigned if the original file name was "Document" and four (Document1 do Document4) files named "Document" were already within the destination folder path.

NOTE

The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~STF::FileName~**, **~STF::Counter~**, or **~STF::FileExt~** with any other component except the Send to Folder eConnector (Process) component, and it must be used with the **Rename** field.

NOTE

You can create and display the counter with the required number of leading spaces and leading zeroes. For example, if the file name is TEST.DOC, and the rename schema is **~STF::FileName~~%03STF::Counter~~STF::FileExt~**, then the resulting file names are TEST001.DOC, TEST002.DOC, and so on.

If the file name is TEST.DOC and the rename schema is **~STF::FileName~~%3STF::Counter~~STF::FileExt~**, the resulting file names are TEST 1.DOC, TEST 2.DOC, and so on (note the two spaces after "TEST").

Field replacement tag name (FRTN). This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special set replacement tag name (SSRTN). This component supports the Date/Time field names shown in the following table:

SSRTN	**Description**
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of the month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"**~STF::%Y~-~STF::%m~**" is replaced with "2004-10"

Troubleshooting tips

Problem	Solution
An error dialog box appears when you attempt to create a folder path.	Make sure that you have not used invalid characters in the folder path definition. Invalid characters are \, /, :, *, ", <, >, and |.
The Path RRTN is not replaced with a path folder.	This situation occurs when you specify a Path, such as Path8, where only six folder-path entries are configured with the Send to Folder component. Check to make sure that the Path number is valid.

Restrictions and limitations

This component has no known restrictions or limitations at this time.

SharePoint Portal v1.0 component

Use the SharePoint Portal v1.0 component to store documents into a centralized, unified interface for enterprise users and highly flexible deployment options.

The only difference between the SharePoint Portal v1.0 *Route* component and the SharePoint Portal v1.0 *Process* component is that the SharePoint Portal v1.0 *Process* component has two additional fields on the **Field Values** tab: **Field Passthrough** and **URL Field Name**. **Field Passthrough** and **URL Field Name** are not available for the Route component.

In an AutoStore process, the SharePoint Portal v1.0 component uses the SharePoint Portal Server (SPS) for document management. Use this component to store files in Microsoft SharePoint Portal Server v1.0.

Feature highlights

You can perform the following tasks by using the SharePoint Portal v1.0 component:

- Provide the general SharePoint Portal information (server, workspace, user name, password, folder path, content source, and workflow).
- Specify a location for storing files.
- Rename files that have duplicate names by using a schema name.
- Check-in files to allow other users to open and update them.
- Assign and change document attributes such as author, title, keywords, description, and categories.
- Repeat file names. The SharePoint Portal v1.0 component appends duplicate filenames with a counter. For example, if the original file name was TEST.TXT, the component rename schema can rename the files to TEST1.TXT, TEST2.TXT, TEST3.TXT, and so on.

Using the SharePoint Portal Route component

In an AutoStore process, the SharePoint Portal v1.0 component is frequently used with the Digital Sender component and the Poll Directory component. In this type of process, the SharePoint Portal v1.0 component picks up files and stores them in the SPS.

For example, if a file is stored in a folder that you want to share with the rest of the company, you can create a process that uses Poll Directory as the Capture component. Save the file that you want to share in a designated Input folder, and then use SharePoint Portal v1.0 as the Route component.

You can use the SharePoint Portal v1.0 component as a Process component only when the Route component in the AutoStore process can accept field values (such as Lotus Notes, open database connectivity [ODBC], and so on).

Configuring the SharePoint Portal v1.0 component

Depending on which Capture component you are using, follow the appropriate procedure to open **SharePoint Portal v1.0 Server** dialog box and configure the SharePoint Portal v1.0 component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the SharePoint Portal v1.0 component.

The following attributes are available in the **SharePoint Portal v1.0 Server** dialog box.

General tab

Use the options on this tab to specify where you want to save incoming files.

- **Server.** When you add the SharePoint Portal v1.0 component, you must specify which server will store the documents that this component processes.
- **Workspace.** The SharePoint Portal v1.0 component uses a workspace concept to provide access to document libraries, content sources, and categories. Select the workspace in which you want to store your information.
- **User Name.** Type a valid user name to log into the SharePoint Portal Server
- **Password.** Type the password that corresponds to the user name to log into the SharePoint Portal Server.
- **Folder Path.** Select a folder path. Folders can have multiple folders within them. The **OK** button is available when you select a folder. A folder might already have other folders in it.

  You can create a folder dynamically by typing a new name for the folder.
- **Content Source.** You can add new content sources to the workspace so that you can gain access to content outside of the document library. Click on "**...**" to view a list of content source from which to select.
- **Workflow.** Select the document library where you want to work. Document libraries can have document libraries within them. Therefore, when you click the browse button (**...**) for **Document Library**, a tree view shows the document libraries hierarchy. You can create infinite document libraries inside document libraries if your computer has enough memory to support them.

  To enable the **OK** button, select a workspace. You can store documents only in the workspace, which is a subordinate document library. After you select the document library, you must then select the Folder Path for the document library.

Document Setting tab

Use the options on this tab to set the document attributes.

- **Author.** Type a valid author name.
- **Title.** Type the title of the document that you are going to store in SharePoint Portal.
- **Keywords.** Type in keywords that will enable improved queries in SharePoint Portal.
- **Description.** Type a description of the document that you are storing in SharePoint Portal.

- **Categories.** Add the **Categories** property to your document to enrich the set of properties that are stored with the document that are captured by search queries.
- **Check in.** Select the **Check in** check box if you want other users to open the file and make changes to it in SharePoint Portal. Until you check the file into SharePoint Portal, other users cannot check the file out. Only one copy of the file can be updated at one time. Users can also add comments to the file when checking the file in.
- **Publish.** Select this check box if you *do not* want the file to be viewed from a Web browser, although it is still stored in SharePoint Portal.
- **Check in Comments.** Before you check in the file, add comments about the changes you made to the file when you checked it out.
- **Rename.** Select this check box to rename the file.
- **Rename Schema.** Select this check box to rename the schema. You can then rename a file that is stored by using the SharePoint Portal v.10 component. If the file name that is being processed uses invalid characters (such as a "\", which can occur when using the PDF Barcode component), you must replace the invalid character with a valid character (such as an "a").

Field Values tab

Use the options on this tab to add, modify, or remove field values.

The only difference between the SharePoint Portal v1.0 *Route* component and the SharePoint Portal v1.0*Process* component is that SharePoint Portal v1.0 *Process* component has two additional fields on the **Field Values** tab: **Field Passthrough** and **URL Field Name**. **Field Passthrough** and **URL Field Name** are not available for the Route component.

- **Add field values.** Click **Add field values** to add new field names and field values to the file that is being stored in the SharePoint Portal v1.0 component. You can add fields such as the title of the file, or any other fields that you require.

  When you click **Add field values**, the **Field Values** dialog box appears. Click the browse button (**…**) to open the **Select Field** dialog box. Select the fields that you want to add to your file. These fields can change, depending on which document library you select. Each document library has its own set of fields. The field types that are currently supported are Text, Number, Boolean, Currency, and Note (that is, multiple lines of text).
- **Modify.** Click **Modify** to modify the field value attributes.
- **Remove.** Click **Remove** to remove a field value.
- **Field Passthrough (for the Process component only).** Select this check box if you want the SharePoint Portal v1.0 Process component to pass all of the fields that it was unable to use on to the next component in the process.
- **URL Field Name (for the Process component only).** Type a URL Field Name string (SPSURL is the default). This string becomes the Reserved Replacement Tag Name (RRTN) for the URL. For example, using the default field name SPSURL, you can reference the URL as "**~SPS::SPSURL~**".

  If instead of SPSURL, you type the string "myurlvalue" for **URL Field Name**, the RRTN would be "**~SPS::myurlvalue~**".

Using Knowledge Package Loader to configure the SharePoint Portal v1.0 component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the SharePoint Portal v1.0 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the SharePoint Portal v1.0 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the SharePoint Portal v1.0 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the SharePoint Portal v1.0 component

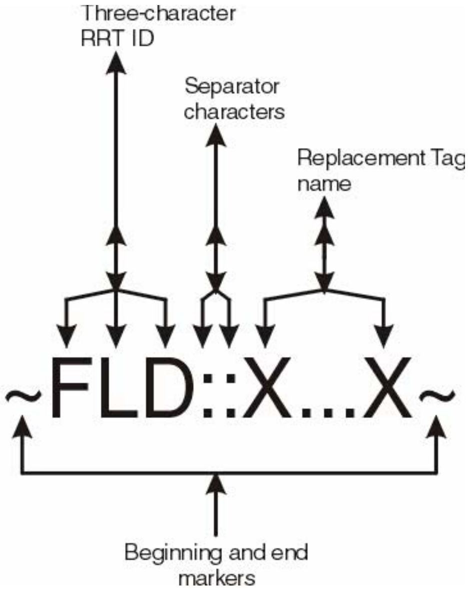
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the SharePoint Portal v1.0 component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and document time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **SPS**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	The original file-extension value.
URLFieldName	The URL of the file that is stored in the SharePoint Portal v1.0 Server (for the Process component only).

The following is an example of the RRTN process:

~SPS::FileName~~SPS::Counter~ is replaced with the filename that is created by using the SharePoint Portal v1.0 component.

~SPS::SPSURL~ is replaced with the URL of the file (where SPSURL is the field name designated for storing URLs). For the Process component only.

Field Replacement Tag Names

This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem description	Solution
The component cannot run.	Make sure that all of the necessary fields have been supplied. These include the server, workspace, user name, password, folder path, content source, and workflow fields.
You cannot gain access to a folder path or a content source by using the browse buttons.	SharePoint Portal Server 2001 client Components (SPSCLIENT.MSI) are mandatory system requirements and must be installed before you can browse in **Folder Path** or **Content Source**. If SPSCLIENT.MSI is installed, make sure that you have typed in a valid user name, password, and server.

Restrictions and limitations

- Do not add two fields that have the same name to **Field Values**.
- The **URL Field Name** cannot have invalid characters such as a "\" or a "." (applies to the SharePoint Portal v1.0 Process component only).

SharePoint Portal 2003 Route component

Use the SharePoint Portal 2003 component to store documents into a centralized, unified interface for enterprise users and highly flexible deployment options.

The only difference between the SharePoint Portal 2003 *Route* component and the SharePoint Portal 2003 *Process* component is that SharePoint Portal 2003 *Process* component has one additional field on the **General** tab: **Pass-through**. **Pass-through** is not available for the Route component.

In an AutoStore process, the SharePoint Portal 2003 component uses Microsoft SharePoint Portal Server 2003 for document management. Use this component to store files in Microsoft SharePoint Portal Server.

The most current version of the SharePoint Portal 2003 component does not require that you install .NET on the computer that is running the AutoStore software.

Feature highlights

You can perform the following tasks by using the SharePoint Portal 2003 component:

- Specify a location for storing files.
- Rename files that have duplicate names by using a schema name.
- Change document attributes.
- Repeat file names. The SharePoint Portal 2003 component appends duplicate filenames with a counter. For example, if the original file name was TEST.TXT, the component rename schema can rename the files to TEST1.TXT, TEST2.TXT, TEST3.TXT, and so on.

Using the SharePoint Portal 2003 Route component

In an AutoStore process, the SharePoint Portal 2003 component is frequently used with the Digital Sender component and the Poll Directory component. In this type of process, the SharePoint Portal 2003 component picks up files and stores them in Microsoft SharePoint Portal Server 2003 .

For example, if a file is stored in a folder that you want to share with the rest of the company, you can create a process that uses Poll Directory as the Capture component. Save the file that you want to share in a designated Input folder, and then use SharePoint Portal 2003 as the Route component.

You can use the SharePoint Portal 2003 component as a Process component only when the Route component in the AutoStore process can accept field values (such as Lotus Notes, open database connectivity [ODBC], and so on).

NOTE To connect to a remote SharePoint Portal server from a client machine, you must run the SP2003WEBSERVICESETUP.MSI file that is located in the WebService Installation folder on the SharePoint server. To do this, locate the WebService Installation folder in the AutoStore directory on the client machine, copy it to the machine that is running SharePoint server, and then run the setup file.

Configuring the SharePoint Portal 2003 component

Depending on which Capture component you are using, follow the appropriate procedure to open the **SharePoint Portal 2003 Server** dialog box and configure the SharePoint Portal 2003 component.

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the SharePoint Portal component.

The following attributes are available in the **SharePoint Portal 2003 Server** dialog box.

General tab

Use the options on this tab to specify where you want to save incoming files.

The only difference between the SharePoint Portal 2003 *Route* component and the SharePoint Portal 2003 *Process* component is that SharePoint Portal 2003 *Process* component has one additional field on the **General** tab: **Pass-through**. **Pass-through** is not available for the Route component.

- **Server.** When you add the SharePoint Portal 2003 component, you must specify which server will store the documents.
- **User name.** Type a valid user name. If the user name that is entered is not an administrator on the SharePoint machine, SharePoint will not allow the user to log in.

NOTE If the user name is not an administrator on the computer on which SharePoint Portal 2003 is installed, the SharePoint Portal 2003 component does not allow the user to log in.

- **Password.** Type a valid password.
- **Domain.** Type the name of the domain on which SharePoint Portal 2003 is running.
- **Site.** Use the Site directory to create various sites to store your documents. Select a site, and then select the document library where you want to work. If you do not specify a site, and then you decide to browse for a document library, the document libraries located in the base of the SharePoint Portal (root directory) are shown. Therefore, a site is not required, although a document library is required.
- **Document Library.** Select the document library where you want to work. Document libraries can have document libraries within them. Therefore, when you click the browse button (**...**) for **Document Library**, a tree view shows the document libraries hierarchy. You can create infinite document libraries inside document libraries if your computer has enough memory to support them.

  To enable the **OK** button, select a work site. You can store documents in only the work site, which is a subordinate document library. After you select the document library, you must then select the Folder Path for the document library.
- **Folder Path.** Select the folder path of the Document Library. Folders can have multiple folders within them. The **OK** button is available when you select a folder. A folder might already have other folders in it.

  You can create a folder dynamically by typing a new name for the folder.

  If a folder path is not specified, the file is stored in the root directory of the document library.
- **Rename.** Select this check box to rename the file.

- **Rename Schema.** Select this check box to rename the schema. You can then rename a file that is stored by using the SharePoint Portal 2003 component.

NOTE

If the **Rename Schema** field is left blank, this field is set to **~SPS::FileName~~SPS::Counter~~SPS::FileExt~**.

For example if the **Rename** check box is selected but the **Rename Schema** field is left blank, and the file "test.doc" was processed through this component, the rename schema will change the name of the file to "test1.doc".

- **Overwrite**. If you do not select the **Rename Schema** check box, this field determines whether a file that already exists in SharePoint Portal 2003 with the same name will be overwritten. If the **Overwrite** check box is not selected, the process will not store a file that has the same name as a file that already exists in the designated folder.
- **Pass-through.** Select this check box when you want the component to pass the document to the next component in the process. This is only valid for the SharePoint Portal 2003 eConnector (Process) component.

Columns tab

Use the options on this tab to add, modify, or remove field values.

- **Add field values.** Click **Add field values** to add new field names and field values to the file that is being stored in the SharePoint Portal 2003 component. You can add fields such as the title of the file, or any other fields that you require.

  When you click **Add field values**, the **Field Values** dialog box appears. Click the browse button (**…**) to open the **Select Field** dialog box. Select the fields that you want to add to your file. These fields can change, depending on which document library you select. Each document library has its own set of fields. All field types are supported.
- **Modify.** Click **Modify** to modify the field value attributes.
- **Remove.** Click **Remove** to remove a field value.

Using Knowledge Package Loader to configure the SharePoint Portal 2003 component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the SharePoint Portal 2003 component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the SharePoint Portal 2003 component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.

5. In the **Name** column, select the SharePoint Portal 2003 component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the SharePoint Portal 2003 component

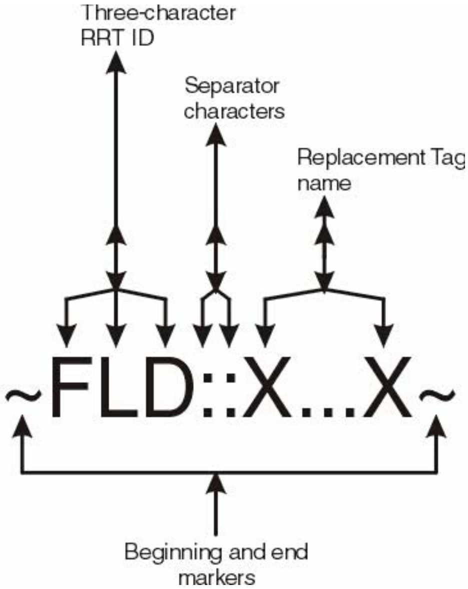
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the SharePoint Portal 2003 component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and document time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **SPS**.

Reserved Replacement Tag Names (RRTN)

The following table describes the RRTN values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	The original file-extension value.
URL	The URL of the file that is stored in the SharePoint Portal server.

The following is an example of the RRTN process:

~SPS::FileName~~SPS::Counter~ is replaced with the filename that is created by using the SharePoint Portal 2003 component.

~SPS::URL~ is replaced with the URL of the file that is stored in Microsoft SharePoint Portal Server 2003.

NOTE

The RRTN values FileName, Counter, and FileExt can only be used with the **Rename** field of this component. You cannot use **~SPS::FileName~**, **~SPS::Counter~**, or **~SPS::FileExt~** in any other component except the SharePoint Portal 2003 component, and it must be used with the **Rename** field.

This rule does not apply to the RRTN value, URL.

When using the SharePoint Portal 2003 Process component, do not use RRT **~SPS::URL~** in a subsequent component's rename schemas. This is because the value of this RRT contains backslashes and if you used it as part of a rename schema, it would create an error because file names cannot contain a backslash. For more information, see the Restrictions and limitations section.

Field Replacement Tag Names

This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem description	Solution
The component cannot run.	To connect to a remote SharePoint Portal server from a client machine, you must run the SP2003WEBSERVICESETUP.MSI file that is located in the WebService Installation folder on the SharePoint server. To do this, locate the WebService Installation folder in the AutoStore directory on the client machine, copy it to the machine that is running SharePoint server, and then run the setup file.
You are unable to locate documents that you stored in the Forms folder.	Avoid storing documents into the Forms folder. Instead, create a new folder and store documents into that newly created folder.

Restrictions and limitations

- Avoid storing documents into the Forms folder. Instead, create a new folder and store documents into that newly created folder.
- For Currency, only valid numbers are acceptable. Do not use **$** or other characters.
- If the **Rename Schema** field is left blank, this field is set to **~SPS::FileName~~SPS::Counter~~SPS::FileExt~**.

- If the user name is not an administrator on the computer on which SharePoint Portal 2003 is installed, the SharePoint Portal 2003 component does not allow the user to log in.
- When using the SharePoint Portal 2003 Process component, do not use RRT **~SPS::URL~** in a subsequent component's rename schemas. For example if your process contains the SharePoint Portal 2003 Process component and a Folder Store Route component, you cannot use the **~SPS::URL~** in the **Rename Schema** field of Folder Store component. This is because the value of this RRT value contains backslashes, and if you chose to use it as part of a rename schema, it would create an error because file names cannot contain backslashes.

Multi Router component

Use the Multi Router component to set the Route component attributes by clicking the **Component Configuration** button to the right of the list. The selected component remains highlighted to indicate that it is active within the Multi Router list. The Multi Router configuration screen lists the Route components that are available for your process. Use the Multi Router component to select one Route component from the list of available components.

Feature highlights

The Multi Router component offers the following features.

- List of all of the Route components that are available to your process.
- Ability to access the configuration attributes of all of the available Route components from one location.

Using the Multi Router component

Use the Multi Router component to use different Route components based on a form or a function key on a digital sender or MFP. The following example illustrates how you can configure the Multi Router component when using the MFP 4100/9000 Capture component.

Case 1: You want to scan documents by using the HP LaserJet 4100mfp. You want one form on the MFP to send the files to a database, and a second form to send files to an FTP server.

Case 1 Solution: For Form1, configure the Multi Router component to store scanned documents in a database by using the Send to Database Route component. For Form2, configure the Multi Router component to send scanned documents to an FTP server by using the FTP Store Route component.

In this example, one AutoStore process defines two distinct treatments for Form1 and Form2 on a single MFP.

Configuring the Multi Router component

Depending on which Capture component you are using, follow the appropriate procedure described here to open the **Multi Router** dialog box and configure the Multi Router component.

The Multi Router component does not require configuration. However, you can use the multi-router component to access the configuration attributes of every other available Route component. To do this, locate the Route component that you want to configure in the **Multi Router** dialog box. Click the button on the right of the component name. This activates that component configuration dialog box, and you can proceed with the configuration as usual.

The selected component will remain highlighted in the **Multi Router** dialog box to indicate that the component configuration dialog box is activated.

Using Knowledge Package Loader to configure the Multi Router component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Multi Router component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Multi Router component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Multi Router component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Multi Router component

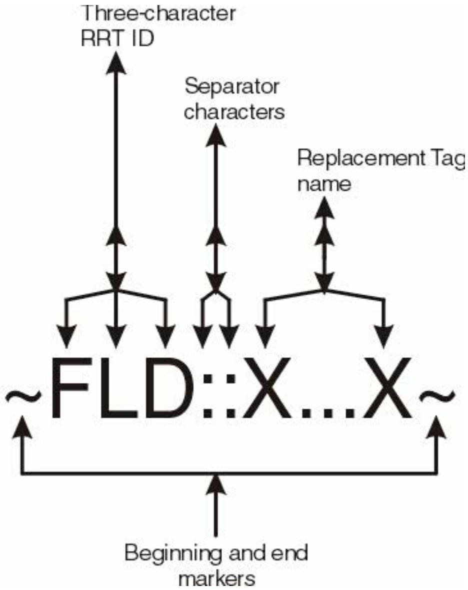
1. Double-click the Digital Sender component.
2. Click the **Digital Senders** tab.
3. Click the **Configure Item** button (lower right).
4. In the **Component Name** window, select the Multi Router component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs.

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

This component has no known restrictions or limitations.

Domino.Doc component

The Domino.Doc component provides scalable content capture server software technology to help you capture your corporate business content into the Domino Document Manager family of products. Domino.Doc improves efficiency through enhanced collaboration and information management. The Domino.Doc Route component delivers scalability, flexibility, and the low cost-of-ownership that is required to support enterprise-wide document, content, and records capture, while serving as a foundation for corporate content capture.

Feature highlights

The Domino.Doc component offers the following features.

- **Content-based routing.** Route documents based on the document content.
- **Multi-server integration.** Write to one or many Domino.Doc servers.
- **Full messaging integration.** Combine the Domino.Doc Route component with other components to take advantage of Notes or Exchange integration features.
- **Integration with libraries, cabinets, and binders.** Capture content at any level and expand your content capture reach by taking advantage of your Domino.Doc hierarchies.
- **Integration with profiles.** Transfer index data from devices, users, documents, or other type of content directly into integrated profiles.

Using the Domino.Doc component

By using the Domino.Doc component, you can perform the following tasks:

- Write documents and index data into back-end libraries.
- Write index data into Domino.Doc and store a URL (or other types of identifiers) pointing to the document location from other systems into Domino.Doc. This method allows you to create a record and link the documents into a central location.
- Use Runtime Replacement Tags (RRTs) to dynamically link multiple components together and allows close integration of extracted content into your Domino.Doc document management system.

Configuring the Domino.Doc component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Domino.Doc component.

Depending on which Capture component you are using, follow the appropriate procedure to open the **Domino.Doc** dialog box and configure the Domino.Doc component.

The following attributes are available in the **Domino.Doc** dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Protocol.** Select the appropriate protocol from the drop-down list.
- **Server.** Type the server name or IP address where Domino.Doc is installed.
- **User Name.** Type the name of the user who has read and write permissions to complete the component configuration.
- **Password.** Type the password for the user designated in the **User Name** attribute.

Document tab

Use the options on this tab to set the following attributes:

- **Library Path.** Type the full path name (which must be DNS-enabled) to the Domino.Doc Library. (For example, the following example shows the server name followed by the path name: **http://DominoServer/domdoc/DominoDoc2Lib.nsf**. In this example, the server name = **DominoServer** and the path to the Domino.Doc Library = **domdoc/DominoDoc2Lib.nsf**.)
- **Cabinet.** Type a Domino.Doc cabinet name. Click "**...**" to browse.
- **Binder.** Type a Domino.Doc binder name. Click "**...**" to browse.
- **Profile.** Select the document Profile where the index values are located. Using the **Field Values** tab, map the index fields into the document **Profile** field. Click "**...**" to browse.
- **Title.** Type the title of the document. The new document is appended to the existing documents.
- **Comment.** Type comments that correspond to the document.

Field Values tab

- **Add.** You can add field name entries and field values to a document. Click the **Add** button to add the new field values to the Domino.Doc component. Type the field name and the corresponding value for the field name.
- **Modify.** Click **Modify** to change the field name value.
- **Remove.** Click **Remove** to remove the field name entry.

Using Knowledge Package Loader to configure the Domino.Doc component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Domino.Doc component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Domino.Doc component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Domino.Doc component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Domino.Doc component

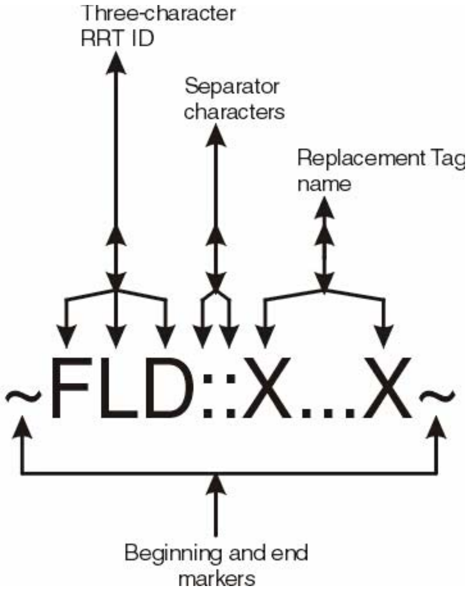
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Domino.Doc component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Domino.Doc component does not generate RRTs; however, all of the attributes can contain RRT strings.

Troubleshooting tips

Problem	Solution
The server name is not a known TCP/IP host name.	Make sure that the script is pointing to the correct Domino server name. Also, make sure that you have a connection to that server on your connection document. Make sure that the server is not referred to by its fully qualified domain name (FQDN), such as MYSERVER.MYCOMPANY.COM. This reference is not registered on the internet as an FQDN, so the client might try to look it up though the DNS, but the client will not find it. To resolve this issue, check the following items: 1. If you have an incorrect input in the FQDN for the server in the address book, remove it for best results. 2. Put in the server name (the Domino server name not the domain name) as the HOSTS file on your machine as an entry for this new Host. EX. 128.202 .205.23 MyDominoServer Make sure that you can ping the server name from the command prompt.

Problem	Solution
Domino/Notes are on the same computer, and you get a TCP/IP error that the <server name> ... is not a known TCP/IP host.	1. Check the preceding solution for Server connection + DNS / host name resolution from the address book. 2. If you made a mistake and pointed to C:\LOTUS\DOMINO during the installation of AutoStore (instead of pointing to the folder C:\LOTUS\NOTES), you must perform some of the following tasks: – Remove AutoStore. – Remove Domino from your system path, for example, C:\LOTUS\DOMINO. – Make sure that you install a full Notes client. – Make sure that you include your Notes executables in the system path, for example C:\LOTUS\NOTES. – Shut down and restart your computer. – Remove the following entry from the NOTES.INI file in your server folder: **EXTMGR_ADDINS=extpwd.** – Delete the following file from the DOMINO folder **C:\LOTUS \DOMINO ..... NEXTPWD.DLL**. – Reinstall AutoStore and point to the correct Notes folder. – Shut down and restart your computer, then load DOMINO, and then start your work.

Restrictions and limitations

The following restrictions apply to installation of this component:

- You must have the Lotus Notes client installed on the same server.
- You must have Domino.Doc desktop installed on the same server.
- Make sure that the Lotus Notes client is in the PATH variable.

NOTE

If you have Lotus Domino Server in the PATH variable, this component will not work.

- Make sure that the password DLL is placed within the Lotus Notes client directory.

Lotus Notes/Domino component

Use the Lotus Notes/Domino component to store information such as text, graphics, scanned images, sound, and digital movies in databases within the Lotus Notes/Domino system. Each database stores two types of information for each document:

- **Document content.** The file that is created by using software programs such as word processors or spreadsheets.
- **Metadata or properties.** The descriptive characteristics such as type, format, title, subject, keywords, and author.

Feature highlights

You can perform the following tasks by using the Lotus Notes/Domino component features.

- Store documents within the Lotus Notes/Domino system.
- Update and add document content or metadata to an existing record in the Lotus Notes/Domino database.

Using the Lotus Notes/Domino component

When you use the Lotus Notes/Domino component, you might be prompted to provide a password, depending on how you installed your software. You can install the AutoStore software on a computer that has the Lotus Note client, the Lotus Notes server, or both. At least one is required. If you install the Lotus Notes client on the computer on which you have installed the AutoStore software, then during the AutoStore configuration you might be prompted for the same password required for the current client user ID before you can connect to the Lotus Notes server. This depends on whether the client user ID even has a password and whether you have already logged in to the Lotus Notes client. However, if you have the Lotus Notes server installed where you are installing the AutoStore software, then you will not be prompted for a password during configuration because the server ID does not have a password.

If the AutoStore server and the Lotus Notes server are installed on the same platform and the Lotus Note client is not installed on that platform, you will not be prompted for a password when configuring either the attachment field or select a form. In some instances, a Lotus Notes client and server exist on the same computer. In this case, you might be prompted for a password if you attempt to connect to a Lotus Notes server and retrieve any database information. For additional information see the Lotus Notes documentation.

The following example is a common usage scenario.

The MFP devices can be configured with a Lotus Notes/Domino button. By pressing this button, images are scanned in and stored in a designated database in the Lotus Notes/Domino system.

Licensing

The Lotus Notes/Domino component does not require any special licensing level.

Configuring the Lotus Notes/Domino component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Lotus Notes/Domino component.

Depending on which Capture component you are using, follow the appropriate procedure to open the **Lotus Notes/Domino** dialog box and configure the Lotus Notes/Domino component.

The following attributes are available in the **Lotus Notes/Domino** dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Server.** (optional) Type the server name or IP address where the Lotus Notes/Domino database is installed.
- **Password.** Type the password for the current user ID that is that is being used by the Lotus Notes/Domino system to log in to a database.
- **Database Name.** Type the name of an existing database.
- **Form Name.** Type the title of a form note in the designated database.
- **Attachment Field.** Type the name of the field within the designated form into which you want to store incoming document content or files.
- **Rename File.** Select this check box to rename the output file based on the **Rename Schema** settings.
- **Schema.** Type a schema name for the output file name. You can use RRTs to dynamically set the value of the schema.
- **Calculate Computed Fields.** Select this check box to initiate a recalculation of field formulas after a record is created or updated. This allows the predesigned field formulas to recalculate all related field forms upon creation of a record using the AutoStore software.
- **Ignore Validation Errors.** Select this check box to enable the AutoStore software to ignore any field validation errors that might be caused. If this check box is not selected, any field validation errors will cause a failure in the creation of a record. Note that this check box is only available when **Calculate Computed Fields** is selected.

Field Values tab

- **Add.** Use this button to assign values to fields on the designated form. Use the search key if you want to search the Lotus Notes/Domino database before creating a new record.

  If the search returns only *one* matching record, the the incoming document content and metadata is added to the existing record.

  If the search returns *more than one* matching record, a new record is created in the database.
- **Modify.** Use this button to change field values on the designated form.
- **Remove.** Use this button to delete field values from the designated form.

Set Field Value On Search

When search keys are used and a matching document is found, then any existing field in the record is replaced with the field values in the current job.

Using Knowledge Package Loader to configure the Lotus Notes/ Domino component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Lotus Notes/Domino component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Lotus Notes/Domino component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Lotus Notes/Domino component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Lotus Notes/Domino component

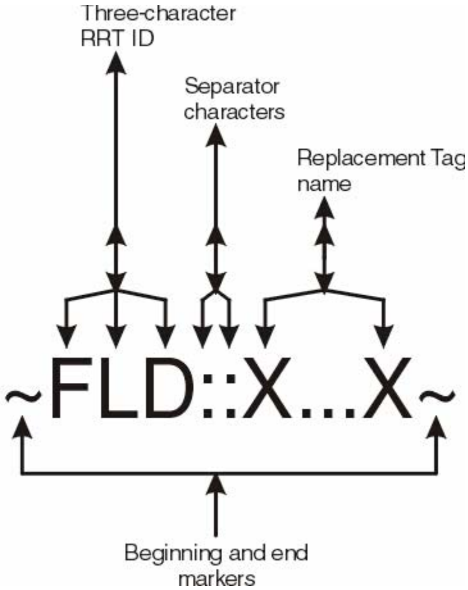
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Lotus Notes/Domino component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT for this component is **LND**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter that is based on the duplicate file names found within a directory. The counter value concatenated with a name provides a unique file name.
FileExt	The original file-extension value.

The following is an example of the RRTN process:

~LND::FileName~~LND::Counter~

The value "Document5" is assigned if the original file name was "Document" and four (Document1 to Document4) files named "Document" were already within the destination folder path.

- **Field Replacement Tag Name** This component does not have any Field Replacement Tag Names (FRTNs) and does not replace field names with metadata values.
- **Special Set Replacement Tag Name** This component supports the Date and Time special set replacement tag names (SSRTNs) shown in the following table.

SSRTN	Description
%a	The abbreviated weekday name
%A	The full weekday name
%b	The abbreviated month name
%B	The full month name
%d	The day of month as a decimal number (01 to 31)
%H	The hour in a 24-hour format (00 to 23)
%I	The hour in a 12-hour format (01 to 12)
%j	The day of the year as a decimal number (001 to 366)
%m	The month as a decimal number (01 to 12)
%M	The minute as a decimal number (00 to 59)
%p	The A.M. or P.M. indicator for a 12-hour or 24-hour clock, as appropriate for the locale
%S	The second as a decimal number (00 to 59)
%U	The week of the year as a decimal number, with Sunday as the first day of the week (00 to 53)
%w	The weekday as a decimal number (0 to 6; Sunday is 0)

SSRTN	Description
%W	The week of the year as a decimal number, with Monday as the first day of the week (00 to 53)
%y	The year without the century, as a decimal number (00 to 99)
%Y	The year with the century, as a decimal number

The following is an example of the SSRTN:

"~LND::%Y~-~LND::%m~" is replaced with "2003-9"

Troubleshooting tips

Problem	Solution
Either of the following two error messages appears: "Unable to find path to server" or "Server is not responding."	Several situations can prevent the Lotus Notes/ Domino component from connecting to a particular server: ● The server is temporarily unavailable. If you cannot connect to the same server using the Lotus Notes client, but you have connected to that server before, this is most likely the reason. ● The server cannot be reached through any of the network ports that are enabled for the current location. If you cannot connect to the same server by using the Lotus Notes client, and you have never connected to that server before, this could be the problem. Contact your Lotus Notes administrator to determine which ports to use. ● The server cannot be reached through the network port used by this component. If you can connect to the same server by using the Lotus Notes client, this is most likely the reason. This happens because the Lotus Notes client uses all of the available network ports when it is trying to reach a server, while this component only uses one port (typically the first port enabled for the current location). To enable a network port, use the following procedure: Reorder the ports so that the port that you need for access to the server is listed first. In the Lotus Notes client, click **File, Tools, User Preferences**, and then click **Ports**, or edit NOTES.INI and reorder the ports in the `"Ports="` entry.

Problem	Solution
You cannot create documents in the database.	Open the Lotus Notes client. Click **File, Database**, and then click **Access Control List** to see the Access Control List for the database where the document is located. To create a document, you must at least have permission rights of Depositor access. You must also select the option **Create documents**.

Restrictions and limitations

- This component must be installed on a computer that has either the Lotus Notes client or the Lotus Notes/Domino server installed.
- The path to the Lotus Notes/Domino directory must be on the system path.
- This component uses the Lotus Notes ID that is specified in the Lotus Notes initialization file (NOTES.INI) when it is attempting to log on to the Lotus Notes/Domino server.

Microsoft Exchange component

Microsoft Exchange is a Route component that can be used to store information of any kind—text, graphics, scanned images, even sound and digital movies—in public folders within the Microsoft Exchange server. Documents can be routed to dynamic folders, a destination form can be selected, and the index field in the form can be set.

Public folders facilitate the exchange of information between groups of people within an organization. Public folders store two kinds of information for each document: document content and metadata, or properties.

- Document content is the file you create by using programs such as word processors or spreadsheets.
- Properties are descriptive characteristics, such as the document type, format, title, subject, keywords, and author.

Feature highlights

The Microsoft Exchange component offers the following features:

- Ability to store documents in public folders within the Microsoft Exchange server.
- Ability to update and add document content or metadata to existing documents in public folders.

Using the Microsoft Exchange component

MFPs can be configured with a Microsoft Exchange button. When you press this button, images are scanned in and then stored in a designated public folder on a Microsoft Exchange server.

Configuring the Microsoft Exchange component

Use static or dynamic values as defined in the Capture component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Microsoft Exchange component.

Depending on which Capture component you are using, follow the appropriate procedure to open the Microsoft Exchange dialog box and configure the Microsoft Exchange component.

The following attributes are available in the **Microsoft Exchange** dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Server.** Type the name of the Microsoft Exchange server.
- **Mailbox.** Type an alias or an account name of a user on the Microsoft Exchange server.

NOTE The Mailbox name is not the profile name.

- **Folder Path.** Click "**…**" to view a list of available folders. Select the folder where you want to place your files.
- **Form Name.** Click "**…**" to view a list of available forms. In the **Select Form** dialog box, click to select the appropriate form, and then click **OK**.
- **Subject.** Type the subject of the document.
- **Text.** Specify the body property of the newly created document.

Field Values tab

- **Add.** Click the **Add** button to assign the new field values to the Microsoft Exchange forms. If a single record (file) is found that matches all search keys, then the document is added to that record. If multiple records are found that match all search keys, then a new record is created and the document is attached to this new document.
- **Modify.** Click the **Modify** button to change the field value attributes.
- **Remove.** Click the **Remove** button to remove a field value.
- **Set Field Value on Search.** When this check box is selected, the matching field values are updated when a record (file) is found that matches all of the search key(s).

Using Knowledge Package Loader to configure the Microsoft Exchange component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Microsoft Exchange component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Microsoft Exchange component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Microsoft Exchange component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Microsoft Exchange component

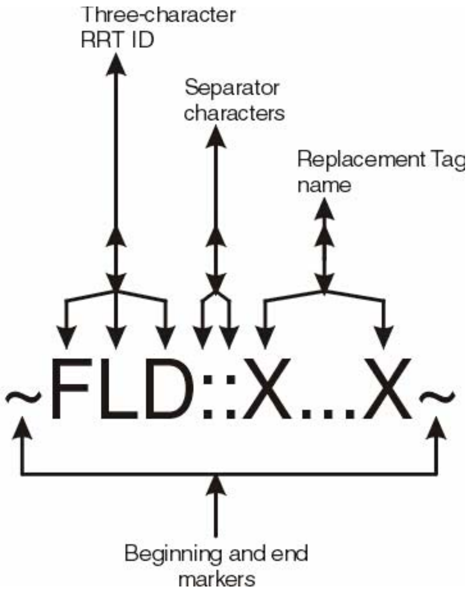
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Microsoft Exchange component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Microsoft Exchange component does not generate RRTs; however, all of the attributes can contain RRT strings.

Troubleshooting tips

Problem	Solution
The form that you want is not available in the public folder.	You might not have permission to use the form, or you might be required to use another form to add information to the public folder. See the folder owner or your administrator. Contacts for a public folder appear on the **Summary** tab. Right-click the appropriate folder, and then click **Properties**.
You cannot create a record in the public folder.	You might not have permission to create items in the folder. See the folder owner or your administrator. Some public folders require that you post information by using a particular form that you might not have set up. See your administrator for information about forms. The item type must match the folder type. For example, you cannot add a task item in a mail folder type. If you want to create a new item of a type that is different from the folder, create the item in the matching folder type, and then move the item into the public folder.
Public folders are not visible in the Folder Path browser.	Outlook and its Collaboration Data Objects option must be installed if you do not have AutoStore installed on the same server on which the Microsoft Exchange component is installed.

Restrictions and limitations

- The Microsoft Exchange component must be installed on the Microsoft Exchange server or on a client computer that has Microsoft Outlook. If this component is installed on a client computer that has Microsoft Outlook, then the feature **Collaboration Data Objects** must be installed.
- The **Log On NT As** account on the AutoStore Service Manager must be the Windows NT account that is associated with the selected exchange mailbox. This is required in order to perform Windows NT authentication and gain access to the Exchange Server.
- If you are storing to Exchange Public folders, do not store in a folder that ends with the <space> character. For example, **My<space>Folder** is acceptable, but **My<space>Folder<space>** will not work.
- If you are running two tasks to two Exchange servers, such as storing documents to two storage areas on different servers, follow these guidelines:
  - Make sure that the same mailbox exists on both servers.
  - Make sure that the mailbox has the same name that the service is using on the AutoStore server.
  - Make sure that you have sufficient rights available for both servers.

VB/J Script component

Use the VB/J Script component as a Process or Route component with the most common scripting languages to manipulate and create your own custom-built capture program. You can write your scripts to gain access to external databases, manipulate internal files, or validate index data fields. You can also use the VB/J Script component to write scripts to gain access to external data sources, look up information that adds value to your capture process, and merge the necessary external data elements.

Scripts offer flexibility that helps you create custom-built Capture components quickly. As files come in, you can run a designated script to alter the files, and then perform various tasks such as saving the files to specific locations.

The VB/J Script component supports VBScript and JScript scripting languages.

Feature highlights

Choose a scripting language to create the VB/J Script component from the following two options:

- JScript
- VBScript

Use this component as a Process or Route component in any process.

The VB/J Script component can be used to process any file type.

Using the VB/J Script component

Use this component to perform the following tasks:

- Gain access to external database files from within your Capture task and validate captured data elements against your internal databases.
- Manipulate and reformat a file in the middle of a process to match your custom needs.
- Apply other program wrappers to files so that you can control the document format, security, and presentation.
- Add, delete, or modify field index data values to the process data space, reduce the amount manual data entry required, and increase your data throughput.
- Inform other users when a specific file type file has been received. You can run a script that sends an e-mail notification each time that particular file type is received.

NOTE When the VB/J Script component is the Route component in an AutoStore process, the script that you choose should provide the Route functionality.

Configuring the VB/J Script component

Use the appropriate procedure to open the **VB/J Script Configuration** dialog box to configure the VB/J Script component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the VB/J Script component.

The following attributes are available in the **VB/JScript Configuration** dialog box.

- **Name.** Type the name of the function that you want to run. For example, you might have many different _OnLoad functions in your script. You must specify which one you want to use. If you specified "Test" as the value of your **Name** field, then the Test_OnLoad function runs.
- **Language.** Select JScript or VBScript from the drop-down list.
- **Script.** Click "**…**" to browse for the script that you want to run. The script must be available in the same directory at runtime.

Using Knowledge Package Loader to configure the VB/J Script component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the VB/J Script component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the VB/J Script component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the VB/J Script component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the VB/J Script component

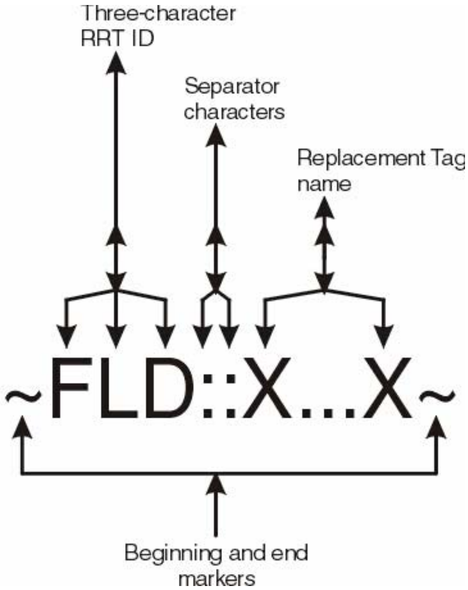
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the VB/J Script component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The VB/J Script component does not generate RRTs. However, all of the attributes can contain RRTs.

For example, when POP3 E-mail is the Capture component, set the "Record Type" parameter to "**~POP::Subject~**" to dynamically set the record type by using the subject field of the e-mail.

Troubleshooting tips

Problem description	**Solution**
Clicking **OK** does not close the dialog box.	Make sure that you have typed a name, selected a language, and specified a script in the configuration dialog box.
A VB/JScript error occurs.	Make sure that the file that you selected as your script is a valid file, and that it has a valid file extension.

Restrictions and limitations

- You must specify a name, language, and script in the **Configuration** dialog box.
- You must specify a valid script file.

Documentum component

Use the Documentum Route component to store information of many kinds, including text, graphics, scanned images, sound, and digital movies, into repositories named docbases within the Documentum system. Each docbase stores two types of information about each document (or type of information stored):

- Document content: The file that you create by using programs such as word processors or spreadsheets
- Metadata or properties: The descriptive characteristics such as the document type, format, and title

Feature highlights

You can perform the following tasks by using the Documentum component:

- Store documents within the Documentum system.
- Instantiate and run Documentum custom components.
- Invoke Documentum custom component to perform additional custom processing on stored documents.
- Create dynamic folder paths. Set the path at runtime, and if the folder path does not exist then it will be created.

Using the Documentum component

The following example is a typical use scenario for the Documentum component:

MFPs can be configured with a Documentum component button. When a user presses the button, images are scanned and then stored into a designated path in the Documentum system.

Configuring the Documentum component

Use the appropriate procedure to open the **Documentum** configuration dialog box to configure the Documentum component.

The following attributes are available in the **Documentum** dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Doc base.** Select the docbase into which you want to store your document content and metadata.
- **User Name.** Type the user name for connecting to the docbase.
- **Password.** Type the password that corresponds to the user name.

- **Domain.** If required, type the Windows NT domain name of the user name that you specified.
- **Path.** Type the location within the docbase into which incoming documents will be stored.
- **Class Name.** Type the programmatic identifier of the custom Documentum component that will be invoked when a document is stored in the system.

Document Settings tab

- **Type.** Type the object type of the document.
- **Format.** Type the format of the document.
- **Name.** Type the name of the document.
- **Title.** Type a name for the document.
- **Subject.** Type the subject of the document.
- **Keywords.** Type keywords that are associated with the document. These will be used for database searches.
- **Authors.** Type the names of the author(s) of the document.

Field Values tab

Use this tab to assign values to custom properties of a given document type.

Using Knowledge Package Loader to configure the Documentum component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Documentum component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Documentum component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Documentum component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Documentum component

1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.

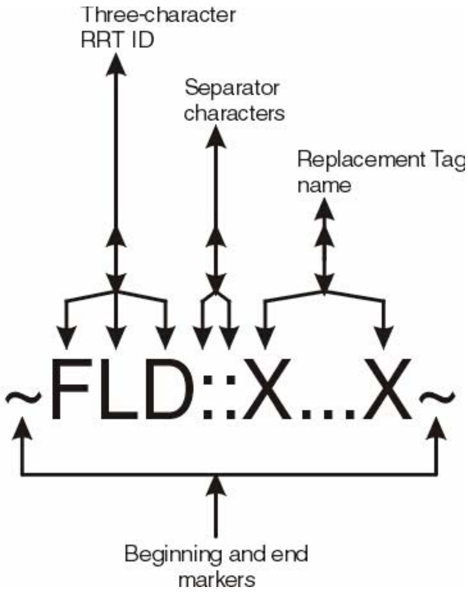
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Documentum component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

This component does not generate RRTs, however all the parameters can contain RRT strings. For example, when POP3 E-mail is the Capture component, the Documentum User Name parameter can be set to "**~POP::To~**".

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

- This component requires the Documentum Foundation Classes (DFC) Version 4.2.1 or later.

FileNET component

Use the FileNET Route component to fully integrate with the FileNET image server. This component provides full capability for mapping runtime documents and index data into libraries, classes, and folders. You can use RRT values to dynamically map all of the available index data that is extracted from the document content or captured from other users, and store it into a FileNET depository.

This component also provides security. Use security integration to set the access level, user, and group assignments.

Feature highlights

You can perform the following tasks by using the FileNET component features:

- Gain full access to the FileNET document libraries.
- Capture various document content types and deposit them into FileNET destination folders.
- Establish correct security settings on the deposited documents.
- Set field values in document classes.
- Use RRT assignments to create dynamic routing.

Using the FileNET component

Use the FileNET component to perform the following tasks:

- Route content directly into FileNET document folders.
- Set field values in document classes.
- Select security user or group assignments, and set the appropriate access level.
- Use Runtime Replacement Tag (RRT) assignments to create dynamic routing.

NOTE The FileNET program must be installed and fully configured on the same server as AutoStore before you can configure the FileNET component. The FileNET component supports only FileNET Version 3.*x*.

Configuring the FileNET component

Use the appropriate procedure to open the **FileNET** configuration dialog box to configure the FileNET component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the FileNET component.

The following attributes are available in the FileNET configuration dialog box.

- **User Network Name.** Select this check box to define the destination attributes such as the document library, class, and folders.
- **User Name.** Type in the user name string for gaining access to the FileNET component. You can assign external values by using RRT tags.
- **Password.** Type in the password for the user name that you defined.
- **Document Library.** Select the name of the document library to which you want the content routed.
- **Document Class.** Select the name of the document class to be used for indexing.
- **Index.** Select this check box to enable the index fields.
- **Destination Folder.** Select the destination folder location name to which you want the documents stored. You can use RRT names within the folder name to make the destination field dynamic.

Field Values tab

You can assign values and deposit captured index data into the document class by using the attributes that are available on the **Field Values** tab.

- **Add.** Click the **Add** button to add a new field mapping definition.
- **Modify.** Select an existing field mapping definition and click the **Modify** button to change the mapping values.
- **Remove.** Select an existing field mapping definition and click the Remove button to delete the mapping assignment.

Security tab

Use the **Security** tab to place documents into the FileNET system and set the appropriate security level for the document access. Note that you can use RRTs for the Name field to create dynamic assignments from the captured index data fields.

- **User.** Select this radio button if the name refers to an individual user (not to a group).
- **Groups.** Select this radio button if the name refers to a group (not to an individual user).
- **Name.** Type the name of the user or group that will be used for the security level assignment.
- **Access Level.** Select an access level from the drop-down list to assign to this user or group, for this particular document within the library.
  - **Add.** After selecting the access level entry definition information, click the **Add** button to create a new access level entry.
  - **Remove.** Select an access level entry definition and click the **Remove** button to delete the entry.

Using Knowledge Package Loader to configure the FileNET component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.

3. In the **Component Name** window, select the FileNET component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the FileNET component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the FileNET component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the FileNET component

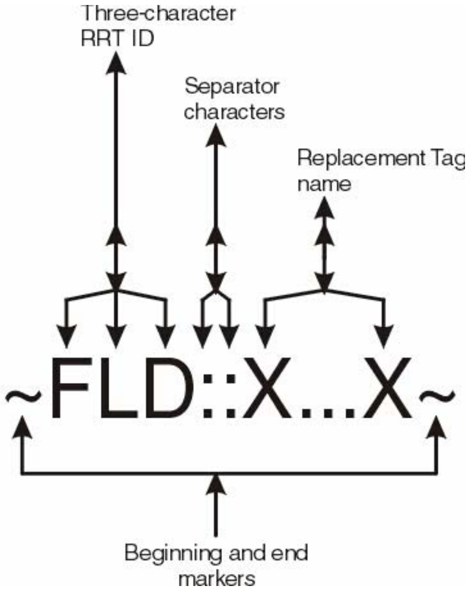
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the FileNET component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The FileNET component does not generate RRTs. However, all of the attributes can contain RRTs.

Troubleshooting tips

Problem	Solution
The documents are not stored into the FileNET destination folders.	Check the Log files and make sure that no errors exist. If you see an error code from this component, ask for an error code definition from the FileNET administrator. Also check the security access-level definition The access level might be restricting you from viewing documents that are deposited into the destination folder.

Restrictions and limitations

- FileNET must be installed on the same server as AutoStore.
- FileNET security must allow access to the server user ID.
- AutoStore supports only FileNET Version 3.*x*.

IXOS Component (updated)

Use the IXOS component to create all of the necessary interface files to store documents and index data correctly into an IXOS program. You can also use the RRT variables with the IXOS component to create custom commands and dynamic-command entries.

The IXOS component fully supports IXOS SAP interfaces. By using these interfaces, you can create command files that have the attributes and variables that are required to write your files into the IXOS application, and then link the documents back to your SAP system.

This component supports the server interface for IXOS eCONServer Batch Import version 5.0.

Feature highlights

Using the features of the IXOS component, you can perform the following tasks:

- Create interface files for storing documents and index data into an IXOS application.
- Use RRT variables to create custom commands and dynamic command entries.
- Use supported SAP interfaces to create command files that have the attributes and variables that are required to write your files into the IXOS application, and then link the files back into your SAP system.

Using the IXOS component

Use the IXOS interface to archive any documents, including document attributes, transparently into IXOS-ARCHIVE. Create the documents within the import directories and then store them in the archive system by using the archive server.

You must transfer the documents, including the attributes, according to a defined form. Use the AutoStore IXOS component to create and transfer documents according to a defined directory on the archive server that is referred to as EXT_DIR. The substructure of this directory is predefined and is described in the *Batch Import with Attributes* guide from IXOS. Within the IXOS component, the path for EXT_DIR is specified when you install it. You must point to the same EXT_DIR subdirectory when you are using the AutoStore IXOS component. You must also ensure that sufficient space is available in EXT_DIR in order to transfer external documents.

Use the IXOS interface to create standard batch import files within the EXT_DIR subdirectory structure. You provide the EXT_DIR root directory on the **EXT_DIR** tab of the IXOS component. The component is responsible for creating the appropriate subdirectories. For details about the EXT_DIR subdirectory structure, see the *Batch Import with Attributes* guide from IXOS.

The IXOS component creates the necessary directory and files under the EXT_DIR subdirectory, including the following files:

- IXATTR file. All of the attributes headers and corresponding fields.
- COMMAND file. All of the commands that are related to the image files and attributes.
- LOG file. An empty log file that is created for the IXOS Batch Import process.
- Image file(s). The captured image file in .TIF or .PDF format.

This interface also supports SAP R3 attributes and commands, as well as Flexible Header structures for free-format creation of IXATTR or COMMAND interface file entries. For details about all of the attributes and command variables see the *Batch Import with Attributes* guide from IXOS.

NOTE You can use RRTs with the IXOS component to create your file entry variables. An example of using RRTs is the use of **~L1B::BARCODE~** within the value field of the IXATTR entry. This type of RRT usage allows all of the IXATTR and COMMAND attributes to be dynamically driven from the metadata values.

Configuring the IXOS component

Use the appropriate procedure to open the **IXOS** configuration dialog box to configure the IXOS component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the IXOS component.

The following attributes are available in the **IXOS** configuration dialog box.

IXAT TR File Builder tab

Use the attributes on this tab to create an attribute record for each job. The entries on this tab are created within the attribute record each time that a new document is placed into one of the IXOS directories. You can select the header and attributes from the predefined list of fields, or you can create attribute entries by using the Flexible Formatting interface.

The predefined headers and attributes provide the standard set (as well as SAP R3 entries) so that SAP integration can be developed for IXOS. Use the Flexible Formatting interface to specify custom attribute lines to further customize the IXOS interface.

You can use Runtime Replacement Tags (RRTs) to create dynamic attribute entries. The following are examples of dynamic attributes that use RRTs:

Type **R3_CLIENT ~L1D::1,1~** as a Flexible Formatting entry.

~L1D::1,1~ represents the first barcode on the first page of the document. Assuming that the barcode value is **XXX**, then the entry in the attribute file will be **R3_CLIENT XXX**.

Use RRTs to create entries within the batch file that link your IXOS batch file entries into other document-related values such as barcodes, user field entries (on devices such as multifunction devices), form field values, zoned OCR field value, or values that are available for any other component. This is a powerful feature that allows organizations to create custom solutions based on their document content-capture requirements.

Command File Builder tab

Use the **Command File Builder** tab to create the correct entries within the command file interface. You can use the following methods to create entries:

- **Predefined headers.** Use this method to select the command entries from a list of defined entries. By using the interface, you can select from a list of available entries and quickly complete your command line. Use the header command entries to select entries such as COMP with Component Type: PDF. The command entry that is created as the result of this entry is COMP 1.pg PDF. The command line entry result is COMP 1.pg PDF <filename>.
- **Flexible Formatting.** Use this method to type your entries and create a complete set based on your custom requirements. Use the flexible command formatting to create entries such as COMP 1.PDF PDF. Use the flexible entries to create custom commands for your IXOS program.

EXT_DIR tab

Use the **EXT_DIR** tab to select a directory where documents and attributes are transferred within a defined directory on the eCONServer archive server. The AutoStore software automatically creates the required subdirectories within this server. You must point to the EXT_DIR as defined by the Batch Import Server interface.

Use RRTs to create dynamic EXT_DIR entries such as **C:\ ~L1D::1,1~\EXT_DIR**, where the barcode component generates **~L1D::1,1~**. The entry represents the first barcode on the first page of the document. This barcode will be replaced with the actual value of the barcode at runtime. For example, if the first barcode on the first page is XXX, then the directory used to write the files will be C:\XXX\EXT_DIR.

Using Knowledge Package Loader to configure the IXOS component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the IXOS component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the IXOS component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the IXOS component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the IXOS component

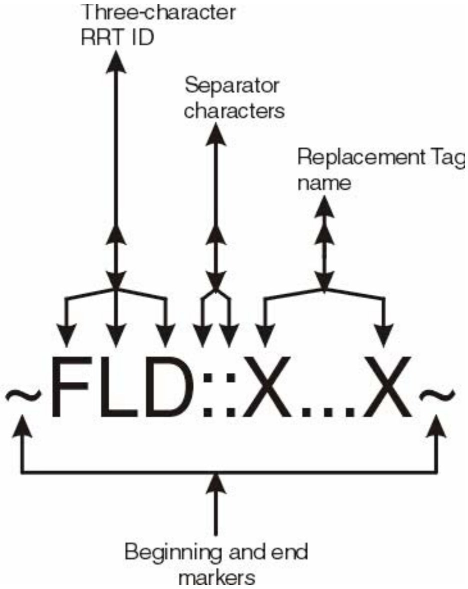
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the IXOS component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The IXOS component does not generate RRTs.

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

This component contains no known restrictions or limitations.

OpenText Livelink component

Use the OpenText Livelink component to capture documents and data into the OpenText Livelink 9.x application. The integration between AutoStore and Livelink enables you to capture documents from a variety of AutoStore Capture components into this document management system.

Log on to the OpenText Livelink component and identify the folder, category, and fields within the component where you want the document to be stored. Set up your AutoStore workflow process to capture content from the specified Capture component, and then store the content in the specified OpenText Livelink folder.

The integration between the AutoStore software and the OpenText Livelink component takes advantage of Livelink features, such as multiple category support, security options, multi-value document and folder naming support.

Features

You can perform the following tasks with the OpenText Livelink component.

- Support various document version control.
- Lock documents for enhanced security.
- Integrate the AutoStore software with the OpenText Livelink component to use the security options and features.
- Support multiple category and folder structure.

Using the OpenText Livelink component

Use the OpenText Livelink component to store any type of content from various sources. This component allows you to easily meet your regulatory and archival requirements for records retention. The following are examples of how the OpenText Livelink component can be used.

Knowledge Management. Capture, process, organize, share, and store valuable information into the OpenText Livelink 9.x document management system.

Direct device connectivity. Allow users to archive important documents by pressing a few buttons on a scanning device. Use the OpenText Livelink component to directly connect various devices such as digital copiers, desktop scanners, production high-speed scanners, desktop files, and other types of files to backend OpenText Livelink 9.x.

Batch Import Server. Use the OpenText Livelink component along with the Poll Directory component to create batch import directories where files read in from various directories can be imported directly into backend OpenText Livelink 9.x.

Uniform Capture Process tools. Create capture business rules that determine how your content is captured into a backend document management system by using the AutoStore process designer tools.

Connect e-mail files to the Livelink document management system. Use POP3 email or SMTP Capture components to connect e-mail content and archive all e-mails within an inbox or all e-mails sent to an SMTP gateway into the OpenText Livelink component for archival, management, or sharing.

Configuring the OpenText Livelink component

The following attributes are available in the **OpenText Livelink** configuration dialog box.

- **Preferences tab.** Configure the settings for the AutoStore process home directory, IP port number, and other administrative directories.

General tab

Use the attributes in this tab to define the connectivity to the OpenText Livelink application.

- **Server.**Type the IP address or the host name of the OpenText Livelink 9.x server.
- **Database.** Type the name of the OpenText Livelink database to which you are connecting. If you specify a null string, the system uses the default database that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
- **Username.** Type a valid OpenText Livelink user name. The processed documents are associated with this user name.
- **Password.** Type a valid password that corresponds with the user name.
- **Port.** Type the port number that the server uses to communicate with the OpenText Livelink server. The port number value must match the port number that is configured on the OpenText Livelink software.
- **Impersonate.** Type the user name of the user that you want to impersonate. You must use the administrator's user name and password.
- **Advanced.** Select this check box to activate the security mechanism that is used by the OpenText Livelink application programmer's interface (LAPI) software to exchange data with the OpenText Livelink server.

The following options are available in the **Advanced** attribute.

Direct Connection. This is the default option. A direct connection does not ensure the confidentiality of the data that is passed over the Internet or any unsecured network. The data messages are passed as unencrypted plain text. If another user intercepts the plain text message, that user can view the content.

Non-secure Tunneling. Non-secure tunneling occurs when a LAPI application exchanges data with an OpenText Livelink server by transmitting unencrypted (plain text) HTTP messages through the Web server that is integrated with an OpenText Livelink server.

- AutoStore sends data in an HTTP request to the Web server that is integrated with the OpenText Livelink server.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the OpenText Livelink server (similar to the functionality when an OpenText Livelink request is made over a socket connection).

- The OpenText Livelink server processes the AutoStore request, generates a response, and then returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which returns the response to AutoStore.

Field name	Description
HTTPUserName	This is the user name that is recognized by the Web server.
HTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	This is the URL to the OpenText Livelink CGI integration process.

When you have selected the **Non-secure Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	Default value	Description of what to use
Server	WebServerHost	This is the name of the computer on which the Web server is running.
Port	80	This is the non-secure port to the Web server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Non-secure Proxy Server Tunneling. Non-secure proxy server tunneling occurs when a LAPI application exchanges data with an OpenText Livelink server by transmitting unencrypted (plain text) HTTP messages through the Web proxy server, which passes messages to the Web server that is integrated with an OpenText Livelink server.

- AutoStore sends data in an HTTP request to the Web proxy server, similar to a Web browser requesting data from a Web server through the Web proxy server.

- The Web proxy server forwards the request to the Web server that is integrated with the OpenText Livelink server.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the OpenText Livelink server, (similar to the functionality when an OpenText Livelink request is made over a socket connection).

- The OpenText Livelink server processes the AutoStore request, generates a response, and then returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which returns the response to AutoStore through the Web proxy server.

Field name	value	Description
HTTPUserName	myHTTPUserName	This is the user that is recognized by the Web server.
HTTPPassword	myHTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	http://host:port/livelink/ livelink.exe	This is the entire URL to the OpenText Livelink CGI integration process where the host is the OpenText Livelink host name and the port is the Web server port number.

When you have selected the **Non-secure Proxy Server Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	Default value	Description of what to use
Server	ProxyServerHost	This is the name of the computer on which the Web proxy server is running.
Port	8080	This is the port to the Web proxy server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Secure Tunneling. This option requires Livelink Secure Connect. You must purchase this product separately, and install Livelink Secure Connect before you can implement secure communications. Livelink Secure Connect includes RSA BSAFE cryptographic and security protocol software from RSA Security, Inc. The data is almost impossible to decipher if it is intercepted, but it is easily converted to plain text by the application that is intended to receive the data.

- AutoStore sends data in a Secure HTTP (HTTPS) request to the Web server that is integrated with the OpenText Livelink server. Browsers use the HTTPS protocol to encrypt user page requests and to decrypt pages that are returned by a Web server.

- The Web server negotiates the SSL connection with AutoStore via an SSL "handshake," and decrypts the application request.

- The OpenText Livelink CGI process acts as a proxy that forwards the request to the Livelink server (similar to the functionality of a typical OpenText Livelink request).

- The OpenText Livelink server process the AutoStore request, generates a response, and returns the response to the OpenText Livelink CGI.

- The OpenText Livelink CGI process forwards the OpenText Livelink server response to the Web server, which encrypts the response and returns it to AutoStore.

Field name	value	Description
HTTPUserName	myHTTPUserName	This is the user that is recognized by the Web server.
HTTPPassword	myHTTPPassword	This is the password that corresponds to the HTTPUserName field.
LivelinkCGI	/livelink/livelink.exe	This is the entire URL to the OpenText Livelink CGI integration process.
CA root certificates		This is a secure LAPI client application that requires the root certificate of the Certificate Authority (CA) from the secure Web server that is integrated with an OpenText Livelink Server to verify the authenticity of the certificate that is passed. In most cases, third-party CAs provide instructions on how to obtain their root certificates on their Web sites (for example, www.verisign.com or www.entrust.com). You can also export some CA root certificates from Microsoft Internet Explorer 5.0 and later.

When you have selected the **Secure Tunneling** option, change the fields on the **General** tab as shown in the following table.

Field name	Default value	Description of what to use
Server	WebServerHost	This is the name of the computer on which the Web server is running.
Port	443	This is the secure port to the Web server.
Database	(null string)	Use the default OpenText Livelink database connection that is assigned to the *dftConnection* variable in the [general] section of the OPENTEXT.INI file.

Field name	Default value	Description of what to use
Username	LivelinkUserName	This is the OpenText Livelink user account that has access to the OpenText Livelink server.
Password	LivelinkPassword	This is the password that corresponds to the LivelinkUserName user account.

Document tab

Use this tab to set the document attributes for the document that is stored in the OpenText Livelink application.

- **Description.** Type a description of the document that you are storing.
- **Folder.** Specify the OpenText Livelink folder destination where you want to store your document. You can dynamically create folders from this location.
- **Lock file.** Select this check box to limit availability of the document. When this check box is selected, only the user who is logged in can gain access to the document. If this check box is not selected, the document is available to all users.
- **Pass-through.** This option is only available on the OpenText Livelink eConnector (Process) component. Select this option when you want the document passed through to the next component in the AutoStore workflow process.
- **Replace.** Select this check box if you want the newly-processed document to replace the existing document.
- **New Version.** Select this check box if you want to store the newly-processed document without overwriting the existing document.
- **Rename.** Type the name of the newly-processed document. See the *Component RRT ID* section for more information.

System tab

Use this tab to gain access to additional node attributes that are configured on the OpenText Livelink server. These node attributes allow you to assign values to the attributes, which are applied to every document in the OpenText Livelink application.

Follow these steps to create new attributes for the OpenText Livelink application:

1. Click **Administer Additional Node Attributes** in the **System Administration** section on the **Livelink Administration** page.
2. Click **Add a New Attribute Link** on the **Administer Additional Node Attributes** page.

3. Type a unique name for the attribute in the **Name** field on the **Add New Attribute** page.
4. Select one of the following attribute types from the **Type** drop-down list:
   - **Text: Field**
   - **Text: Popup**
   - **Text: Multiline**
   - **Flag: Checkbox**
   - **Date: Field**
   - **Date: Popup**
   - **Number: Text field**
   - **Number: Popup**

Categories tab

Use this tab to create and remove association between the processed document and the categories that are defined in the OpenText Livelink application.

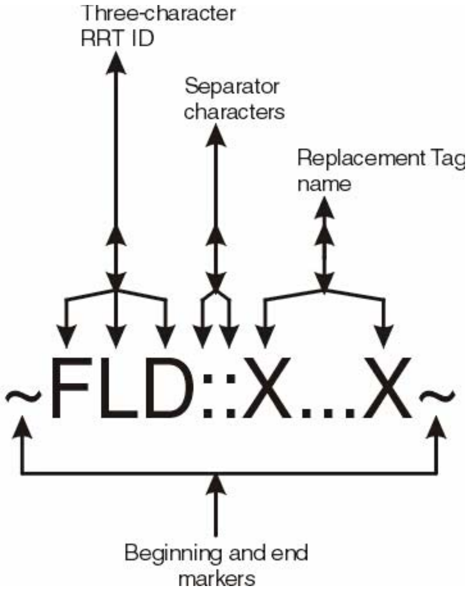
- **Select.** Click **Select** to view a list of all of the categories that are defined in the Enterprise workspace. You can associate your document with multiple categories. Select a category that you want to associate your document with and populate the field values if appropriate.
- **Remove.** Click **Remove** to remove the association between a document and a category.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **OTX**.

Reserved replacement tag name (RRTN). The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	This is the original file name value.
Counter	This is an incremental counter that is based on the duplicate file names that are found within a directory. The counter value is concatenated with a name to provide a unique file name.
FileExt	This is the original file extension value.

The following is an example of an RRTN:

~OTX::FileName~~OTX::Counter is replaced by the value "Document5" if the original file name was "Document" and up to four (Document1 to Document4) files already existed within the destination folder path.

Field replacement tag name (FRTN). This component does not support FRTNs and replacement of field names with metadata values.

Special set replacement tag name (SSRTN). This component does not support SSRTNs.

Restrictions and limitations

- You cannot create categories through the OpenText Livelink component. A category must be created via the Livelink application configuration first.
- You cannot create fields through the OpenText Livelink component. Fields must be created via the Livelink application configuration first.

Troubleshooting

If no categories are available, either your user name and password are incorrect or you have not created any categories on the Livelink Enterprise Server.

Hummingbird (5.x) Route component

Use the Hummingbird Route component to store documents into a document-management system where records can be identified and searched for by using database queries. The only difference between the Hummingbird Process and Route components is that the Pass-through field option on the **General** tab is only available with the eConnector (Process) component.

NOTE The Hummingbird component runs on a client computer that has access to the Hummingbird document-management server.

Feature highlights

You can perform the following tasks by using the features that the Hummingbird component provides.

- Select the library that you want to use.
- Select the profile that you want to use.

This component accepts any document or image file as an input type.

NOTE You must type a valid user name and password in order to populate the **Profile** field.

Using the Hummingbird component

Use the Hummingbird component to store any type of content from a variety of sources. You can meet your regulatory and archival requirements by using the Hummingbird component. The following examples show some common uses of the Hummingbird component.

- **Corporate content repository.** Use the Hummingbird component to protect valuable corporate information. If you want to store documents onto a server where other users (all of whom have permission rights to the files) can view the document, save these documents onto the server, and then give the other users permission to open and use the files.
- **Direct device connectivity.** Use the Hummingbird component with a scanning device to archive important documents. You can directly connect the Hummingbird component to devices such as digital copiers, desktop scanners, production high-speed scanners, desktop files, and other types of files, and then to the back-end Hummingbird document-management system.
- **Batch import server.** Use the Hummingbird component with the Poll Directory component to create batch import directories, where files that have been read from various directories can be imported directly into the back-end Hummingbird document-management system.

- **Uniform capture process tools.** You can use the Hummingbird component process-design tools to create business rules that dictate how your corporate content is captured into back-end document-management systems.
- **Connect e-mail files to the back-end Hummingbird document-management system.** Use the POP3 E-mail component or the SMTP Capture component to connect e-mail content and to archive all of the e-mails within an inbox, or all of the e-mails that were sent to an SMTP gateway, in your back-end Hummingbird document-management system.

Licensing the Hummingbird component

Three types of licenses are available for this component: Evaluation, Licensed, and Expired.

- **Evaluation.** A 30-day fully functional component is available at the first installation.
- **Licensed.** The fully licensed component provides full capabilities.

NOTE Becoming fully licensed requires a Hummingbird client, which is not included.

- **Expired.** After the evaluation period, unlicensed components expire and will not process documents.

Configuring the Hummingbird component

Use the appropriate procedure to open the **Hummingbird** configuration dialog box to configure the Hummingbird component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Hummingbird component.

The following attributes are available in the **Hummingbird** configuration dialog box.

General tab

Use the options on this tab to set the general user attributes of the document-management server.

- **User Name.** Type the user name for connecting to the Hummingbird document-management server. This user must be able to gain access to various libraries and deposit documents. You can use RRT values to create a dynamic login user name based on the user who will be sending documents.
- **Password.** Type the password for the user name that you defined.
- **Domain.** Type the Windows domain name.
- **Logon Type.** Select the appropriate logon type, which is specified by the document-management server. Select from the following logon types that are available in the drop-down list: Library, Microsoft Network, Network Bindery, or Network NDS.

- **Library.** Select the library to which the user will connect. Note that the user must have the appropriate security permissions.

  When you select a library, two other text boxes become active, depending on which library you select. For example, if your library is the default library, then the text boxes remain inactive. However, if you select the Legal, Finance, or Government library, then the text boxes become active, so that you can set the other required features for those specific libraries.

  The Finance library requires **Account** and **Department** fields. The Legal library requires **Client** and **Matter** fields. The Government library requires **Organization** and **Department** fields.
- **Profile.** Select the profile that you want to use to store the document.
- **Typist.** Type the name of another user in the Hummingbird library. The value of this field *must* be another user in the Hummingbird library. The Last Edited By field in the document is set to this value.
- **Impersonate.** Type the name of the user that you want to impersonate. The user name that you specify here becomes the author of the document, instead of the user name that is specified in the **User Name** field. If you leave this field blank, the author is the user name that is specified in the **User Name** field. When you specify a user name in the Impersonate field, that author also receives security permissions.
- **Rename.** Select this check box and specify the schema that is to be used. If you do not select this check box, and the library does not take duplicate names, the document will fail to be stored if another document that has the same name already exists in that library.
- **Pass-through.** Enable this option to pass the document to the next component in the process.

NOTE

The **Pass-through** field is only available with the Hummingbird eConnector (Process) component.

Document tab

Use the options on this tab to set the following attributes.

- **Secure Document.** Select this check box so that only the user who is logged in and the designated **Typist** can see or make changes to the document. Otherwise, any user can gain access to the document.
- **Folder.** Select a folder to which you want to add the document. If you leave this field blank, the document is added to the root directory.
- **Add.** After specifying file extensions, click the **Add** button to add the extension and relate it to a file type. You can specify many file extensions and types.
- **Modify.** Select this button to change an existing file extension.
- **Remove.** Select an existing file extension and click the **Remove** button to delete the file extension.

Fields tab

The user creates the attributes listed here in the DM designer.

- **Fields.** Type the name of the field.
- **ID.** Specify the supported object type. Select from the following types: Edit, ComboBox, CheckBox, MediumEdit, RadioGroup, MultiEdit, and WideEdit.
- **Type.** Specify a supported type field. These types are supported: String, Date, Time, and Integer.
- **Required.** Type `Yes` if this is a required field, or `No` if this is not a required field.
- **Value.** Type the field value.

Using Knowledge Package Loader to configure the Hummingbird component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Hummingbird component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Hummingbird component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Hummingbird component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Hummingbird component

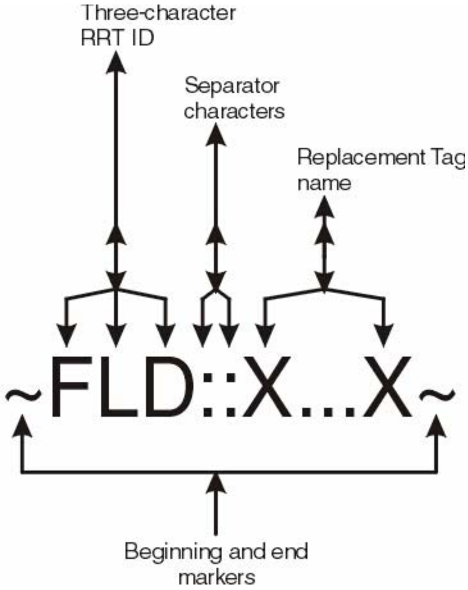
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Hummingbird component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for this component is **HUM**.

The following table describes the Reserved Replacement Tag Name (RRTN) values for this component.

Name	Description
FileName	The original file name value.
Counter	An incremental counter based on the duplicate file names found within a directory. The counter value concatenated with a name provides a unique file name.
ID	The ID number of the document that is stored in the Hummingbird component.
Library	The value of the library.
Domain	The domain name that is used when logging in to the Hummingbird component.

The following is an example of an RRTN:

~HUM::FileName~~HUM::Counter~

Replaced with the value "Document5" if the original file name was "Document" and up to four (Document1 to Document4) files already existed within the destination folder path.

NOTE

The RRTN values FileName and Counter can only be used with the **Rename** field of this component. You cannot use **~HUM::FileName~** or **~HUM::Counter~** in any other components except Hummingbird, and the RRTN must be used with the **Rename** field.

This rule *does not* apply to the RRTN values ID, Library, and Domain.

Field Replacement Tag Names (FRTN)

This component does not have any FRTNs.

Special Set Replacement Tag Names (SSRTN)

This component does not have any SSRTNs.

Troubleshooting tips

Problem	Solution
The **Document Type** box is not populated.	Make sure that you have provided a valid user name, password, and domain, and the appropriate logon type and library.

Restrictions and limitations

- You might have to add one library at a time.
- You cannot select multiple libraries.
- If you store a file into the DM server without specifying a corresponding extension in the file extension list, the file will not be saved correctly.

ApplicationXtender (Route)component

Use the ApplicationXtender Route component to store documents into the Documentum ApplicationXtender, a content-management software package that brings large amounts of data online in a cost-effective manner.

The ApplicationXtender component provides robust and comprehensive security to protect sensitive business information. You can limit access to information within applications, which can further protect confidential information. In addition, user privilege security is provided so that users can be restricted from performing specific functions.

The ApplicationXtender component provides comprehensive electronic file management capabilities and supports a wide range of electronic content.

Feature highlights

Perform the following tasks by using the ApplicationXtender component features.

- Secure your user name and password.
- Select an application into which you want to store your documents.
- Set the title, subject, author, keywords, and comments that are associated with your documents.
- Place a document into a queue for further processing.
- Set the field values for the application that you selected.

The ApplicationXtender component is typically used with the Digital Sender or Poll Directory Capture components. Capture the files by using a Capture component, and then process them by using the ApplicationXtender component. This component can process any file type.

Using ApplicationXtender

This is an example of how to use the ApplicationXtender component:

The Digital Sender device captures content data once and routes it to the SharePoint Portal server. Create your process with the Digital Sender Capture component. Use the ApplicationXtender Process component to manipulate the captured data within the ApplicationXtender repository and save the data to the SharePoint Portal server. This AutoStore process provides a uniform capture capability across the enterprise and storage to the application or media that you select.

Licensing component

Three types of licenses are available for this component:

- **Evaluation.** A 30-day fully functional component is available upon first installation.
- **Licensed.** The fully licensed component provides full capabilities indefinitely.
- **Expired.** After the evaluation period, unlicensed components expire without any further processing.

Configuring the ApplicationXtender component

Use the appropriate procedure to open the **ApplicationXtender** configuration dialog box to configure the ApplicationXtender component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the ApplicationXtender component.

Use the following options to configure the attributes for this component.

General tab

Before using the ApplicationXtender component to manage a document file, you must specify a valid DSN, user name, and password.

- **DSN.** Type the data source name to which you want to connect.
- **User Name.** Type the appropriate user name for the selected data source. If the ApplicationXtender component is using the Windows NT security provider, you must precede the user name with a domain name and a slash symbol. For example, documentation\rfrost indicates that “rfrost” is a user on the “documentation” domain.
- **Password.** Type the password that corresponds to the user name that you specified.
- **Application.** An application is the highest level of organization in the ApplicationXtender component. Use the application for storing and retrieving documents.

  Every time you store a document in an application, you must type index information for that particular document into the index fields. The ApplicationXtender component stores the index information in a database so that you can search it later to retrieve documents.
- **Pass-through.** This option is only available on the ApplicationXtender eConnector (Process) component. If the **Pass-through** option is activated, the documents are passed on to the next component in the process.

Attributes tab

If the computer on which the ApplicationXtender component is installed has been configured to allow searching by open document-management API (ODMA) attributes, you can search for documents by title, subject, author, keywords, comments, and the user name under which the document was created.

- **Title.** Type the title of the document.
- **Subject.** Type the subject of the document.
- **Author.** Type the name of the author of the document name.
- **Keywords.** Type the keywords for the document. Make sure that the keywords are separated by a comma.
- **Comments.** Type comments about the document.

Queue tab

You can place a document into a queue for further processing. The processing queues are used for batch OCR, full-text indexing, and printing.

- **Submit document to queue.** Select whether or not a document should be placed into a queue. If you decide to place the document into a queue, you need to provide a queue name.
- **Queue Name.** Select the name of the queue that you want to place the document into, after storing it into the ApplicationXtender component.
- **Description.** Type a description of the job in which the document will be submitted to a specified queue.

Field Values tab

When a document is added to an ApplicationXtender application, specify data for each of the index fields in the application. Each index field that is defined will be used to hold descriptive information about the documents that are stored in the application.

- **Field.** Type the field name.
- **Type.** Specify the field type. You can select from Text, Integer, Date, Boolean, and so on.
- **Required.** Specify whether or not the field is required. The process cannot be saved until the required fields are assigned a value.
- **Value.** Type the value of the field.

Using Knowledge Package Loader to configure the ApplicationXtender component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the ApplicationXtender component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the ApplicationXtender component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the ApplicationXtender component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the ApplicationXtender component

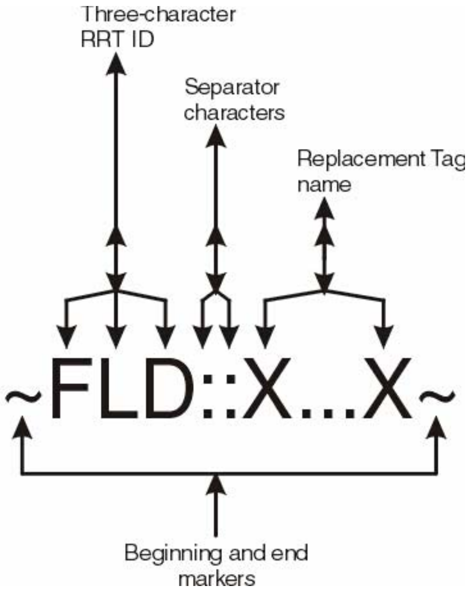
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the ApplicationXtender component.
5. Click **Configure**.

Runtime replacement tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an .XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the .XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The RRT ID for the ApplicationXtender Process component is APX.

NOTE The ApplicationXtender Route component does not generate RRTs.

Troubleshooting tips

Problem description	Solution
No applications appear in the Application field drop-down list.	Make sure that the user name and password are valid. Make sure that you created an application that has corresponding user rights in the Application Generator.
An error message appears indicating that you have to specify a value for all required fields.	Make sure that all of the fields that have the value YES under **Required** have a field value.

Restrictions and limitations

- This component is compatible with ApplicationXtender Version 4.x.

iManage component

Use the iManage Route component to manage repositories of millions of documents for thousands of users. You can use the iManage component to provide searching, document check in and check out, version control, document profiling, and complex security functionality.

Organizations can use the iManage component to make their iManage systems more useful by sending documents directly from different sources, such as digital senders, multifunctional devices, scanners, or POP3-E-mail, into iManage, which offers enough flexibility to design this process for specific business requirements.

Feature highlights

You can perform the following tasks by using the features provided by the iManage component.

- Store a document and all of the profile information that is associated with that document, including the name, description, default security, type, class, author, operator, and custom profile fields that are specific to the iManage implementation, such as client and matter.
- Specify users and groups and the document permissions that they should be allowed.
- Import any kind of document into iManage as long as you set up valid type and class associations.

Using the iManage component

Use the iManage component to extend the capabilities of an iManage system and to provide users with an alternative way to store documents. This component, when used in combination with other Capture and Process components, allows an organization to implement very complex workflows, involving either interactive participation of the users or an unattended environment.

Use this component to categorize documents by using index data that was obtained from barcode cover sheets, or indexing stations, OneStep use, or from multifunctional devices that support indexing.

Use the POP3 E-mail Capture component with the iManage Route component in a workflow scenario where you want to send e-mail documents to a general-purpose public folder in iManage. Assume the business process to be that you are only interested in knowing who sent the document along with a brief description of the document. The following information is a step-by-step description of what needs to take place:

1. You write an e-mail using your iManage User Name in the subject field, and a brief description of the document in the e-mail body. Attach the document that you want to send to iManage. After the e-mail is prepared, send the e-mail to an address that is defined through the POP3 E-mail component.
2. POP3 captures the e-mail and proceeds to detach the document and create RRTs based on the e-mail headers that iManage can use.
3. The system proceeds to route the document to the iManage component. The administrator has configured iManage with fixed values for each one of the required parameters, except for the Author field and Document Description. Instead of using fixed values, RRT from the POP3 component, **~POP::To~** (which represents the "To" field of the e-mail received) has been assigned to the Author and **~POP::Body~** (which represents the message body of the e-mail) has been assigned to the Description.

Configuring the iManage component

Use the appropriate procedure to open the **iManage** configuration dialog box to configure the iManage component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the iManage component.

The following attributes are available in the **iManage** configuration dialog box:

General tab

Use the General tab to configure the parameters for connecting to the iManage library where the documents will be stored.

- **Server.** Type the iManage DeskSite server name or host IP address.
- **Database.** Type the iManage database name.
- **User Name.** Type the iManage user name. The name must have the appropriate permissions to be able to create documents and publish them in the specified database/ server.
- **Operator.** Type the name of the operator of the document. This user must have read and write permissions, and can set the document permissions.
- **Author.** Type the name of the author of the document. This user has maximum security permissions.

Document tab

- **Name.** Type a short name for the document. The name can be up to 16 characters.
- **Description.** Type a description of the document. The description can be up to 254 characters.

- **Default Security.** These are the general security settings for this folder. Three possible default security settings are available.

  When the default security setting is set to **PUBLIC**, all users that have access to the library can view and edit the document and its profile information except those users and groups of users to whom you specifically deny access on the **Security** configuration tab.

  When the default security setting is set to **PRIVATE**, no users have access to the document except the Author and Operator who are named on the **General** configuration tab and users and groups to whom you specifically grant access on the **Security** configuration tab.

  When the default security setting is set to **VIEW**, all users have read-only access to the document, except the Author and Operator who are named on the **General** configuration tab and those users and groups to whom you specifically grant full access, read-write access, or no access on the **Security** configuration tab.

- **File.** Type the file attributes. The file type is typically based on the program that was used to create the file. Document class defines the default settings for the document.

  - **Add.** After specifying file extension, document type and class, click the **Add** button to add the extension.

  - **Modify.** Click this button to change an existing file extension, type, or class association.

  - **Remove.** Select an existing file extension and click the **Remove** button to delete the file extension.

  For example, if the process is receiving scanned images (in .TIFF or .PDF format) from an MFP, click **Add** to create two entries. One entry is for documents with the .TIFF file extension and another entry is for documents with the .PDF file extension. You can also dynamically set these values at runtime by using scripting in AutoStore, or by using a combination of the different Runtime Replacement Tags that other components provide.

Field Values Tab

On this tab, specify the name and values for the custom fields that you want to associate with the stored documents.

- **Add.** Click the **Add** button to enter the name of the custom category in the **Name** box, and its corresponding value in the **Value** box.

NOTE When you type the name of a custom field, the name must match the internal name that iManage uses to refer to this field, which can be **Custom1** through **Custom30**.

  If you are entering values through custom fields, you must provide valid values that come from the custom field lookups. These fields also have a subcategory associated with them. For Custom1, the subcategory is Custom2; for Custom29, the subcategory is Custom30. For the subcategory fields, you must provide valid values from the Custom2 and Custom30 lookups associated to the Custom2 and Custom30 fields. Note that if you are entering values for custom fields Custom2 and Custom30, you must provide valid values for fields Custom1 and Custom29, respectively.

  If you are entering values for custom fields Custom3 to Custom12 you must provide valid values that come from the Custom3 to Custom14 lookups.

  In general, you must make sure that valid values are provided for all of the required fields, based on the Class that will be associated with the document. Failure to do so will cause the document to be rejected because all of the required fields were not found.

  As with other parameters in the configuration, you can enter a combination of fixed tags and RRTs in the **Field Values** tab, or for more complex workflows scripting can be used to populate these values at runtime.

- **Modify.** Select the field that you want to change and click the **Modify** button.
- **Remove.** Select the user or group that you want to delete and click the **Remove** button.

Security tab

Use this tab to specify which users or groups can have access to the document that you are storing. You can also identify the access level that users can have.

- **Add Group.** Click this button to add a group of users to the list. Type a valid name of an iManage group, and then select an appropriate access level from the drop-down list. You can use RRTs to dynamically generate this level, if the final value that the component uses is valid. The four access levels are as follows:
  - **Read-only access.** The user can view the document in read-only format, but cannot make changes and save them to the library as the same document.
  - **Read-write access.** The user has full view and edit privileges to the document. Users with read-write access can change profile information for the document, but they cannot change the author or operator, or edit access privileges. Only the author or operator and those to whom full access has been granted can edit these fields.
  - **Full access.** The user has full view and edit privileges to the document, and can change the security settings for the document. The author and operator have full access, and they can grant full access to others.
  - **No Access.** The user cannot view the document. If a user has no access to a document, the document does not show up in any search of the library that a user performs. If the document is in a public project, users who have no access to the document do not see the document when they click on that project icon.
- **Add User.** Click this button to add a user to the list. Type a valid name of an iManage user, and then select an appropriate access level from the drop-down list. You can use RRTs to dynamically generate this level, as long as the final value used by the component is valid. The four access levels are as follows:
  - **Read-only access.** The user can view the document in read-only format, but cannot make changes and save them to the library as the same document.
  - **Read-write access.** The user has full view and edit privileges to the document. Users with read-write access can change profile information for the document, but they cannot change the author or operator, or edit access privileges. Only the author or operator and those to whom have been granted full access can edit these fields.
  - **Full access.** The user has full view and edit privileges to the document, plus they can change the security settings for the document. The author and operator have full access, and they can grant full access to others.
  - **No Access.** The user cannot access the document. If a user has no access to a document, the document does not show up in any search of the library that a user performs. If the document is in a public project, users who have no access to the document do not see the document when they click on that project icon.

Using Knowledge Package Loader to configure the iManage component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the iManage component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the iManage component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the iManage component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the iManage component

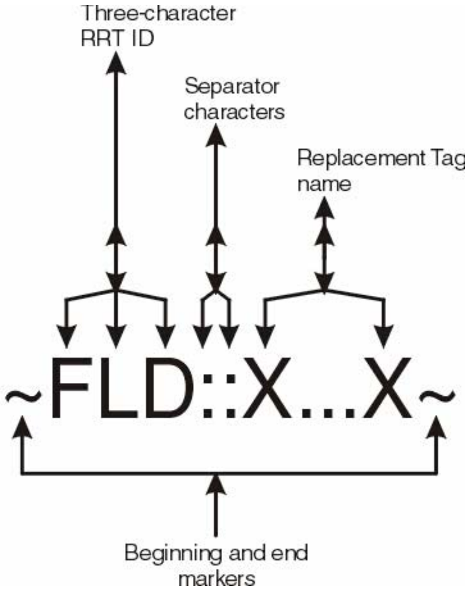
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the iManage component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.

Segment name	Description
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The iManage component does not generate RRTs; however, all of the attributes can contain RRTs. For example, when the POP3 E-mail component is the Capture component, the Author parameter can be set to **~POP::To~**, which represents the "To" field of the e-mail received .

Set the Description parameter to **~POP::Body~**, to represent that the message body of the e-mail has been assigned to the Description. In this example, a document exists in the iManage system, and you can identify the subject and the sender.

Troubleshooting tips

Troubleshooting tips are currently not available.

Restrictions and limitations

- The iManage component does not support version control. Any document that is added is added as a new document. You can add a new version of the document afterward by using any other iManage interface that has version control features.
- The Subclass category cannot be specified through the iManage component. Do not try to set it through the **Field Values** tab, because the Subclass category is not considered a custom category.

Send to Database Batch component

Use the Send to Database Batch component to process batch write files or data into open database connectivity (ODBC) tables such as Microsoft Access or Visual Fox Pro.

Use this component to directly update structured query language (SQL) tables with data elements, or to create a document archival system.

NOTE The ODBC data source must be set up correctly before you configure the Send to Database Batch component.

Feature highlights

Using the features of the Send to Database Batch component, you can perform the following tasks:

- Process batch write files or data into open database connectivity (ODBC) tables.
- Directly update structured query language (SQL) tables with data elements.
- Create a document archival system.

Using the Send to Database Batch component

This information is currently not available.

Configuring the Send to Database Batch component

Use the appropriate procedure to open the **Send to Database Batch** configuration dialog box to configure the Send to Database Batch component.

Use static or dynamic values as defined in the source component Runtime Replacement Tags (RRTs) to set the case-sensitive attributes for the Send to Database Batch component.

The following attributes are available in the **Send to Database Batch** configuration dialog box.

General tab

Use the options on this tab to set the following attributes:

- **Data Source.** Click "**…**" for a list of available data sources. Click to select a data source. This is a required field.
- **User Name.** (optional) Type the user name for logging in to the Data Source.
- **Password.** (optional) Type the password for logging in to the Data Source.
- **Table Name.** Type the table name to which you want to connect. This is a required field. If you receive an error, you have an incorrect data source, an incorrect user name, or an incorrect password.

Field Mapping tab

Use the options on this tab to set the following attributes:

- **Field Delimiter.** Type the field delimiter that is used in the .CSV file.
- **Direct Mapping.** Select this check box to automatically map database field names from the column names in the first row of the .CSV file. Note that the first row of the .CSV file must be reserved for column names. If you do not select this check box, you must manually map the batch fields.
- **DB Fields.** This attribute shows the database fields from the selected table in the database.
- **Type.** This attribute shows the database field type.
- **Batch Fields.** Use the automatically mapped or manually type the .CSV file header field names from the first row of the .CSV file.

Using Knowledge Package Loader to configure the Send to Database Batch component

1. Double-click the Knowledge Package Loader component.
2. Click the **Components** tab.
3. In the **Component Name** window, select the Send to Database Batch component.
4. Click **Configure**.

Using MFP 4100/9000 to configure the Send to Database Batch component

1. Double-click the MFP 4100/9000 component.
2. Click the **MFP Menu** tab.
3. Click the **Add Form** button.
4. Click the **Components** tab.
5. In the **Name** column, select the Send to Database Batch component.
6. Click **…** in the **C** column.

Using Digital Sender to configure the Send to Database Batch component

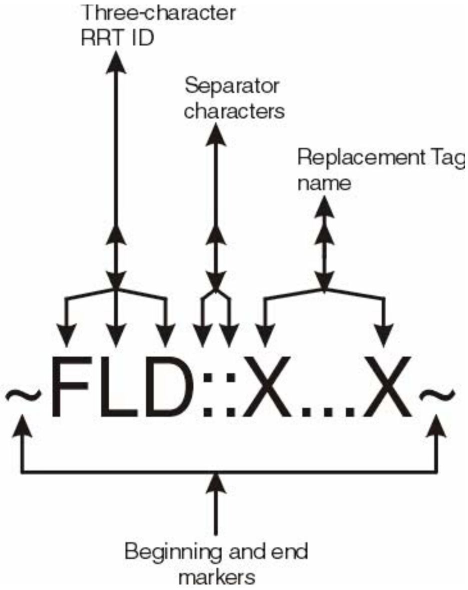
1. Double-click the Digital Sender component.
2. Click the **Digital Sender** tab.
3. Click the **Configure Components** button (lower right).
4. In the **Component Name** window, select the Send to Database Batch component.
5. Click **Configure**.

Runtime Replacement Tags (RRTs)

Runtime Replacement Tags (RRTs) are tags within the configuration file that are replaced by metadata values when the file runs. At runtime (when a component is activated within the program), each component is responsible for searching all items within the configuration memory structure and replacing the assigned RRTs with actual metadata values. The RRTs were formerly known as Field Value References (FVRs).

RRT naming conventions

The following figure illustrates the RRT definition. Each part of the RRT creates a standard field-tag replacement that can be expanded to all of the components that are designed and developed within the component framework.



NOTE

RRTs are case sensitive. You must use the correct case for all RRTs. Do not introduce space characters between the RRT segments.

The following table provides information for each segment of the RRT.

Segment name	Description
Markers	Markers are special characters that designate the beginning and the end of each RRT. The "~" (tilde character) is used to mark each end. Note that you cannot use the same character anywhere within the other parts of the RRT. "~" cannot appear as a value of a special field with this new RRT design.
RRT ID	The RRT ID is a special three-character value that identifies the component that both owns the RRT and is responsible for replacing the tags with the actual values.

Segment name	Description
Separators	Double colon "::" characters are used to separate the component RRT ID from the actual Replacement Tag Name. The double colon is reserved and cannot be used within other parts of the RRT. An example of an unacceptable use of "::" is "~MYC::Invoice::Number~", in which the double colon is repeated within the replacement tag name.
Replacement Tag Name	The Replacement Tag Name (RTN) can be the name of the metadata that is collected from a content source (such as the device), or the metadata that is generated by the component itself (such as an XML data field). The following types of RTN are currently supported: ● **Reserved (RRTN).** Each component can create and maintain a set of reserved RTN or RRTN. Within the boundaries of each component, reserved names have special meaning. Each component defines a list of RRTNs. ● **Field (FRTN).** Some components that support field names allow the fields to be referenced by using the special enclosure character "%". The following is an example of a field name used within the RRT: ~MYC::%Invoice Number%~ In this example, a field named "Invoice Number" within the MYC component can be replaced with the value of the Invoice Number field. ● **Special Sets (SSRTN).** These are special sets of characters that provide useful information about the process, such as document date and time. Not all components support the SSRTN values. See the Help file for each component for information about SSRTN support.

How does it work?

When you create a process that consists of a series of components, the components are completed in order from left to right (Capture to Route). Each component searches all of the configuration entries and replaces all of its own RRTs with actual metadata values. For example, the XML Knowledge Package Loader component searches for RRTs that start with "~ASX::", and replaces them with actual metadata values.

The order of components within a process is significant. Because components are completed in a sequence that begins with Capture and ends with Route, RRTs that are owned by components that appear later in a process cannot be placed within components that appear earlier in a process. For example, placing Folder Store component RRTs within a Capture component (such as Poll Directory) is invalid, and the RRTs will not be replaced at runtime.

Component RRT ID

The Send to Database component does not generate RRTs.

Troubleshooting tips

This information is currently not available.

Restrictions and limitations

This component contains no known restrictions or limitations.

8 AutoStore templates

Using AutoStore templates is a simple method of saving a frequently used set of configuration parameters. If you want to create a configuration file that reuses components, processes, processing steps, and attributes, use a template that includes this information. After a template is created, you can always copy this template to a new configuration file by selecting it when the AutoStore Process Designer presents the list of templates.

Creating a template from a new configuration file

Creating your own templates is a two-step process. First, create a blank configuration file that you believe should become a template for future use. Next, save the file as a template.

1. Open the AutoStore Process Designer. Click **Start**, select **Programs**, select **AutoStore**, and then click **AutoStore Process Designer**.
2. On the Process Designer toolbar, click **File**, and then click **New**.
3. Select **Blank Process** in the **New** dialog box, and then click **OK**.
4. The **Process Information** dialog box appears. This dialog box contains the process attributes. Type a name for the new process in the **Process Name** field.
5. Click **OK**.
6. Drag and drop a Capture component from the component tray to the blank process in the right pane.
7. Click the **Process** tab under the component tray to see the available Process components. Skip to step 9 if you are not adding any Process components.
8. Drag and drop one or more Process components from the component tray to the process in the right pane.
9. Click the **Route** tab in the component tray to see the available Route components.
10. Drag and drop one Route component from the component tray to the end of the process in the right pane.
11. On the AutoStore Process Designer toolbar, click **Save**.
12. In the **Save As** dialog box, type a name for the process, and then click **Save**.
13. In the AutoStore Process Designer, click **File**.
14. Click **Save As**.
15. Type a file name for the new template.
16. Select **CTF AutoStore Template** as the file type, and then click **Save**.

NOTE

You must set up and save your template files in the default template directory. In the AutoStore Process Designer, click **Tools**, and then click **Options**.

Saving an existing configuration file as a template

You can save an existing configuration file as a template.

1. Click **File**.
2. Click **Save As**.
3. Type the configuration file name.
4. Use the drop-down list to change the **Type** to **CTF AutoStore Template**, and then click **Save**.

Saving a process as a template

You can also save the AutoStore process that you are currently working on (not the whole configuration file) as a template. That is, the process might contain a Capture component, any number of Process components, and a Route component, but no process attributes.

1. While working on the process in the AutoStore Process Designer, click **Processes**.
2. Click **Save As Template**.
3. Type the template file name, and then click **Save**.

NOTE All template files must be stored in the AutoStore Process Designer default template directory.

Modifying a template

To modify an existing process template, open the template in the AutoStore Process Designer, and then follow these instructions to overwrite the existing template file.

1. Click **File**.
2. Click **Save As**.
3. Select the template that you want to replace.
4. Select **CTF AutoStore Template** as the file type, and then click **Save**.

9 Secured transport

AutoStore uses encryption to provide secured data transport.

Secured and non-secured transmission

As the distributed imaging process becomes more common in the marketplace, the issues of securing images in transmission needs to be addressed. A number of legislative actions require privacy of information for medical, financial, and banking systems. The secured transformation of images along with related data insures that sensitive information is handled correctly and keeps potential perpetrators from gaining access to it.

The science of concealing information is known as cryptography. Cryptography techniques have been used by governments for centuries to keep military transmissions secret. As the information age matures, the same techniques are now used to secure personal information as it travels across the Internet.

Alternatives in securing transmission

Two major techniques exist for encrypting information: symmetric and asymmetric cryptography. The cryptography systems that make use of the same key for both encryption and decryption are called symmetric cryptography, and the systems that use public and private keys are referred to as asymmetric.

The advantage of using symmetric cryptosystems is speed. When a large amount of information needs to be encrypted, the symmetric system has distinct advantages because its higher speed provides a more efficient means of encryption. The asymmetric system advantage is key management, because the encryption is performed with the "public key" and decryption is done by using the "private key." The key management is easy, but the speed is an issue with asymmetric cryptosystems.

Because of the speed consideration, most cryptosystems that secure large amounts of data use symmetric cryptography techniques and then pass the symmetric keys by using asymmetric cryptography. AutoStore has implemented the first stage of security by implementing a number of symmetric encryption techniques for both images and data transmission. The second stage, which uses Secure Sockets Layer (SSL) and Private Communication Technology (PCT) negotiation, an encryption algorithm for systematic session key exchange, is planned for future releases of the software.

Both symmetric and asymmetric cryptosystems are regulated by the United States government. It is illegal to export certain type of cryptosystems from the United States. AutoStore makes use of "approved for export" technology that is provided within the Windows operating system worldwide.

AutoStore security features

AutoStore makes use of the latest version of CryptoAPI from Microsoft to produce a secure interface for transmission of images and associated data.

The Crypto API 40-bit RC2 and RC4 with Microsoft RSA default CSP have been approved for export outside the United States and are available worldwide. These encryption methods are now offered as part of the secured interface. This technology is currently available in AutoStore, and it provides a strong and sophisticated cryptographic technique.

The user or system administrator is responsible for managing and maintaining the keys or passwords independently of the software.

The following are important points to remember when you use the AutoStore secured interface:

- The encryption key must match at both AutoStore locations.
- The system administrator is responsible for installing the encryption key and protecting it.
- The encrypted messages can be transmitted to a directory by using FTP.
- Each site can be implemented using a separate and distinct encryption key.
- Each FTP directory can have a separate security key assigned to it.
- All images that are transmitted to a secured FTP directory must be encrypted with the same key.
- All encrypted messages that cannot be decrypted are placed in the Rejected directory.
- If encryption password is lost, you must re-scan the images. (You cannot recover a lost password or encrypted data or image files without the correct encryption key.)

Turning on security features

Within the AutoStore architecture, you can implement "filters" for processing images and data attributes. These filters are applied at various stages of the workflow, depending on the configuration parameters and the workflow setup.

The security module is available as a processing filter and is applied before transmission. The receiving AutoStore applies the same filter as soon as files are received to decrypt the files before processing them.

To turn on the security, just select encryption in your AutoStore Process Designer interface and type your security key. The security is automatically applied the next time you run AutoStore with that configuration file. AutoStore makes it simple to turn security on. Just click encrypt or decrypt and then specify your encryption key.

To indicate that a directory contains secured data or image files, turn the security on at the receiving end, and type the decryption key. (This key must match the encryption key.) The security is automatically turned on for you.

Managing keys

The Administrator sets the encryption key the first time that encryption is turned on. The key can remain the same for the lifetime of the AutoStore installation and does not expire. It *cannot* be revoked. To ensure extra security, some organizations might implement periodic updates to the encryption key. Any such procedure must follow the instructions outlined in this guide to ensure that correct and synchronized key changes are implemented.

Changing the keys

The administrator can change the encryption key at any time. The device at the receiving end recognizes the changes in the encryption key and starts placing any files that are processed with newly changed key in the rejection directory (unprocessed). After all of the files that are encrypted with the previous key have been processed, the administrator should change the receiving AutoStore key to the new encryption key and move any files from the rejected files directory to the inbox for processing. This procedure should be repeated for all of the secured remote sites that have encryption keys; the encryption key at transmitting sites must be changed first, and then the same key must be installed at the central receiving site.

Lost keys

Any images that have been encrypted with a lost or forgotten key cannot be recovered. Hewlett-Packard does not hold any master key to any encryption technology that is used within AutoStore. All files that have been encrypted with a forgotten or lost key must be re-scanned.

Secured e-mail by using AutoSafe

AutoSafe is a simple executable file that registers itself for the .CRY file extension. If you receive an e-mail attachment with .CRY (short for Cryptographic) extension, you need to use AutoSafe to decrypt the file.

AutoSafe prompts for the secret decryption key and the encryption algorithm. This information must be supplied by the trusted encryption partner that has sent the files.

Secured application storage using AutoSafe

By using the AutoStore Encryption component, you can store secured and encrypted files in a document management program. No one other than individuals that have access to the private decrypting key and encryption algorithm can open these files. Use this technique to add an extra level of encryption to your document management system.

Finding more information

For more information about security, see the following sources:

- For information about Microsoft Internet Security Framework, go to www.microsoft.com/security/default.asp.
- For information about public-key cryptography, go to www.rsa.com.

The following documents explain aspects of security and encryption.

- CCITT, Recommendation X.509, "The Directory-Authentication Framework." Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva, 1989.
- Schneier, Bruce. Applied Cryptography, 2d ed. New York: John Wiley & Sons, 1996.

10 Problem solving

The following troubleshooting information is organized to help you resolve problems that might occur when you use AutoStore. If you need additional information about a problem, see the following sources:

- The user guide or service manual for the digital sender or MFP that you are using with AutoStore.
- The MFP online Help system. HP MFPs feature an online Help system that provides instructions for resolving common problems. To gain access to Help, press **?** on the control panel.
- AutoStore Process Designer online Help. To gain access to Help, on the AutoStore Process Designer toolbar, click **Help**, and then click **Contents**.
- AutoStore Status Monitor messages. To gain access to the Status Monitor, click **Start**, select **Programs**, select **AutoStore**, and then click **AutoStore Status Monitor**.

Tips for avoiding problems

Use the following tips and suggestions to implement uniform AutoStore practices and avoid problems while using AutoStore.

- *Where do you store your configuration files?* Keep all of your configuration files organized in one directory on the server. If you have a large number of configuration files, use subdirectories to organize them further. Remember that each AutoStore 2002 server can be started by using one configuration file.
- *How do you name configuration files?* Define and use a file-naming convention to name your configuration files.
- *Where do you run the AutoStore Process Designer (APD)?* You must run the APD on the computer where AutoStore is installed. The APD looks for registered components and displays these components in the Component tray. If AutoStore is not installed, APD returns an error.
- *Where do you store your templates?* Always store your template under the Template subdirectory. Remember that a template contains all of the processing parameters, including process names and attributes.
- *How do you set up the Inbox, Working, Success, and Failure directories?* The best way to create your set of directories is to create them under a directory with the same name as your process directory. For example, if you have a process called “SPS Store,” then create a directory called “SPS Store” and create your Inbox, Working, Success, and Failure directories in “SPS Store” directory.

Resolving common issues

Cause	Solution
Transfer button in the AutoStore License Manager is not turned on.	Make sure that the component that is selected is licensed and is not an evaluation copy.
The **Update** button in the AutoStore License Manager is not turned on.	Make sure that the component that is selected is not licensed. Otherwise, you cannot obtain a license.
AutoStore software stops functioning during evaluation.	You must purchase a license key within 30 days of installation.
You are unable to install the AutoStore software.	Make sure that your system meets the following minimum requirements: Windows 2000 with Service Pack 4, or Windows XP Professional; Intel Pentium III, 600 MHz or faster; Support database engine client, such as Microsoft Exchange 5.5 with Service Pack 3.0, Lotus Notes 4.6.x, or any other supported client program; 512 MB RAM; 512 MB available hard disk space.
Packages compacted by pkzip for MS DOS and gzip do not work with Remote Chai Loader	Do not use packages that are compacted by using pkzip for DOS and gzip.
An error message appears.	Make sure that the RRT is in the correct uppercase and lowercase format. RRTs are case-sensitive.
A menu item does not appear in the menu of the MFP. Only the HP AutoStore main menu is available, and no options appear in the menu. If you attempt to stop the AutoStore service in the AutoStore Service Manager, and the message `Unable to stop the service` appears, the AutoStore Service probably did not start. It is also possible that the AutoStore service has not updated the menu yet. The default menu refresh interval is 60 minutes.	It is possible that an SMTP service is running on the workstation or server on which AutoStore is installed. Check the Service Control Manager (click **Start**, select **Control Panel**, select **Administrative Options**, and then click **Services**) and stop and service that is labeled **SMTP**. The default menu refresh interval can be changed in the embedded Web server **Autostore Configuration** section. In the **Configuring HP AutoStore on each MFP** section, set the refresh interval to one minute. However, before you put the MFP into a production environment, change this interval to a longer period.

Cause	Solution
After you send a document, the file does not appear in the destination folder because the destination folder settings are incorrect.	Make sure that you look for the file in the folder that is specified in the Folder Store configuration in the AutoStore Process Designer. If the folder destination is on a remote server share, make sure that the folder is not restricted to Read only. The folder can either be open for anonymous Writes, or the account used to start the AutoStore service must have Write access to this folder. Note that the account used to start the AutoStore service must also have administrative rights on the AutoStore server. The account must also have the **Logon as a Service** attribute selected in the **Local Security Policy**. Look in the Rejected Files Folder, as defined in the AutoStore Process Designer on the **Preferences** tab. If the file you scanned by using the MFP appears in the Rejected Files Folder, the folder path is not correctly defined in the Folder Store configuration.
Each document that you scan by using the MFP appears in the destination folder, but it overwrites any previously sent scanned job.	This behavior is correct if the **Overwrite Existing File** option is selected in the Folder Store configuration. However, you can modify the menu so that you can name files dynamically before you scan documents. In addition, you can also insert a counter in the filename (for example, image**1**.TIF, image**2**.TIF, and so on) so that previously scanned jobs are not overwritten. Use FVR Strings to rename files. For more information, see the Folder Store Help file, or see Runtime replacement tags (RRTs).

Resolving issues with loading .JAR files

If you receive an error message or experience other problems when loading Chai .JAR files, follow the appropriate procedure described in this section to resolve the issue.

Cause	**Solution**
The MFP is out of available RAM. The .JAR load fails before completing, and the following message appears: `Exception caught: hp.chaiserver.PkgLoaderException: Security Violation: Unable to verify jar: http://15.98.155.198/autostore/hp_laserjet_dynamicmenus_30a_1.1.jar`	Turn off the MFP, and then turn it on again. Reload the .JAR file. If this does not resolve the issue, load an additional memory module in the MFP.
You attempted to load the associated .XML file instead of the .JAR file. The .JAR load fails immediately and status page shows the following message: `Unable to authorize: Security policy check failed`	Make sure to load the .JAR file rather than the associated .XML file.
You attempted to load a .JAR file and **Directory browsing** has not been selected in the **Edit Properties** dialog for the Web share. There might also be issues with the Web share. The .JAR file load fails immediately and the following message appears on the status page: `Processing http://15.98.155.198/ChaiJars/s_hp_autostore_4100_9000_2010.jar Verification exception: java.util.zip.ZipException: bad directory signature ffffffffffffff Abort loading of: http://15.98.155.198/ChaiJars/s_hp_autostore_4100_9000_2010.jar hp.chaiserver.PkgLoaderException: java.lang.NullPointerException`	Make sure that **Directory browsing** is selected in the **Edit Properties** dialog for the Web share. If this does not resolve the issue, see your network administrator for information about troubleshooting the Web share.
The Remote Chai Loader (RCL) times out.	The Web share is not configured correctly. If the RCL times out, turn off the MFP, and then turn it on again. Next, open the embedded Web server again, and then attempt to load the .JAR file.
The embedded Web server is not available after you installed the .JAR file, turned off the MFP, and then turned it on again. This occurs when the DNS server (configured for the Jetdirect card) is not configured for a valid DNS server. This can also occur if the network configuration of the Jetdirect card changes after the .JAR files have been installed. (For example, changing from a static IP address for the Jetdirect card to a DHCP address will create this issue.)	Make sure that the DNS server and the Jetdirect card are configured correctly.

Cause	Solution
Because the .JAR files did not load correctly, no additional items appear in the menu on the 4100mfp or 9000mfp.	Open the RCL dialog box in the embedded Web server. (In a Web browser's address line, type `http://<YOURDEVICEADDRESS>/hp/device/this.loader`.) The .JAR files should appear in the **Installed Packages** section. If the .JAR files have not been installed, repeat the installation of the .JAR files. You might need to perform a disk initialization to ensure that the .JAR files load correctly.

Using the About dialog box

The **About** dialog box contains the following information:

- The product logo
- The product name
- Copyright information

To open the **About** dialog box, click **?** on the AutoStore Process Designer toolbar.

AutoStore messages and error codes

The AutoStore Status Monitor shows real-time status messages that are associated with all of the active processes on a server that is running AutoStore. Monitoring these status messages helps you produce the appropriate result for a given process. It can also help predict and identify the sources of any potential system problems. Status Monitor messages contain the following information.

- **Type:** The type of status message. This can be one of the following types:
  - **Error:** Error types indicate significant problems that you should know about. Error events usually indicate a loss of functionality or data.
  - **Warning:** Warning types indicate problems that are not immediately significant, but that might indicate conditions that could cause future problems.
  - **Information:** Information types indicate successful operations.
- **Message:** The message text associated with an event.
- **Time:** The time of the event on the server.

Glossary

AutoStore Process

An AutoStore process is a series of components connected together for processing data or images.

Capture component

A Capture component is type of component that is responsible for reading data or images to a process (for example, Poll Directory).

Component

Components directly handle the manipulation of the data or images (for example, Pro OCR, SharePoint Portal, Microsoft Exchange, or Domino.Doc). Components are combined to create a process.

Configuration file

A series of AutoStore processes *and process attributes* are stored within a configuration file. The configuration file has a .CFG file extension.

Function keys

Function keys are located on the control panel on the HP LaserJet 9100C Digital Sender. By using the HP Address Book Manager, you can assign a name for an AutoStore process to a function key. This name appears on the digital sender control panel for you to select. The assigned names appear under **Customized Function Keys** in the the Address Book Manager main window. Up to 11 first-level function keys can be customized in this way. For each function key, you can configure up to 512 destinations.

Knowledge object

The knowledge object is the memory record that contains the file and metadata for an AutoStore job. The knowledge object for a specific AutoStore job is created when the job is captured by the Capture component and contains all of the metadata associated with the AutoStore job. The knowledge object can be thought of as the "payload" that contains the file and all related metadata. Note that all of the fields that are loaded into the knowledge object are automatically saved into the Route component record when the destination data storage has a matching field name.

Mapping component

This type of component maps the processing attributes of other components to its own internal properties. For example, the Digital Sender component maps the attributes for OCR, form recognition, SharePoint Portal, and so on to each digital sender, function key, and sub-function key.

Options

By using options, you can set the environment options for the AutoStore Process Designer.

Process component

A Process component is a type of component that is responsible for manipulating the content of the data or images (image processing, form recognition, and so on).

Process template

A process template is a type of configuration file that provides basic tools for shaping a final configuration file. Templates can contain one or more processes, process attributes and settings, and/or component attributes and settings.

Route component

A Route component is a type of component responsible for storing objects to a final storage location (for example, SharePoint Portal).

Routing slip

A routing slip refers to the component attributes defined for configuring a function key on a digital sender (or a form on an MFP). The routing slips are defined within the Digital Sender or MFP component attributes.

Toolbar

Using the AutoStore toolbars to execute commands within AutoStore. To customize the toolbar, click **Tools**, and then click **Customize**.

Index

www.hp.com